The Importance of Being Formal

Martin Henz

July 10, 2013

Predicate Logic

1 Need for Richer Language

Propositional logic can easily handle simple declarative statements such as:

> Student Peter Lim enrolled in CS3234.

Propositional logic can also handle combinations of such statements such as:

> Student Peter Lim enrolled in Tutorial 1, *and* student Julie Bradshaw is enrolled in Tutorial 2.

However, statements involving formulations such as *"there exists..."* or *"every..."* or *"among..."* are difficult to express in propositional logic. A statement of the form

> *Every* student is younger than *some* instructor.

talks about concepts such as

- being a student,
- being an instructor, and
- being younger than somebody else

These are *properties* of elements of a *set* of objects. We express them in predicate logic using *predicates*.

Example 1. *The statement*

> Every *student is younger than* some *instructor.*

is expressed using the following predicates.

- *S: For example,* $S(andy)$ *could denote that Andy is a student.*

- *I: For example, $I(paul)$ could denote that Paul is an instructor.*
- *Y: For example, $Y(andy, paul)$ could denote that Andy is younger than Paul.*

A practical problem arises when such predicates are used to express statements such as

> *Every* student is younger than *some* instructor.

How do we express *"every student"*? We need *variables* that can stand for constant values, and a *quantifier* symbol that denotes *"every"*. Using variables and quantifiers, we can write:

$$\forall x(S(x) \to (\exists y(I(y) \land Y(x, y)))).$$

Literally: For every x, if x is a student, then there is some y such that y is an instructor and x is younger than y.

Example 2. *Consider the following statement.*

> *Not all birds can fly.*

Using the following predicates,

$B(x)$**:** *x is a bird*

$F(x)$**:** *x can fly*

we can express the sentence as follows:

$$\neg(\forall x(B(x) \to F(x)))$$

Example 3. *Consider the following statement.*

> *Every girl is younger than her mother.*

Using the following predicates,

$G(x)$**:** *x is a girl*

$M(x, y)$**:** *x is y's mother*

$Y(x, y)$**:** *x is younger than y*

we can express the sentence as follows:

$$\forall x \forall y(G(x) \land M(y, x) \to Y(x, y))$$

Note that in the previous example, the variable y is only introduced to denote the mother of x. If everyone has exactly one mother, the predicate $M(y, x)$ is a function, when read from right to left.

We introduce a function symbol m that can be applied to variables and constants as in

$$\forall x(G(x) \to Y(x, m(x)))$$

Example 4. *Consider the following statement.*

Andy and Paul have the same maternal grandmother.

Without function symbols, we would have to write

$$\forall x \forall y \forall u \forall v (M(x,y) \wedge M(y, \mathit{andy}) \wedge M(u,v) \wedge M(v, \mathit{paul}) \rightarrow x = u)$$

However, with the function symbol m*, we can simply write:*

$$m(m(\mathit{andy})) = m(m(\mathit{paul}))$$

2 Predicate Logic as a Formal Language

At any point in time, we want to describe the features of a particular "world", using predicates, functions, and constants. Thus, we introduce for this world:

- a set of predicate symbols $\mathcal{P}$
- a set of function symbols $\mathcal{F}$

Every function symbol in $\mathcal{F}$ and predicate symbol in $\mathcal{P}$ comes with a fixed arity, denoting the number of arguments the symbol can take. Function symbols with arity 0 are called *constants*.

Definition 1. *The set of* terms *in predicate logic is given by the BNF:*

$$t ::= \ x \mid c \mid f(t, \ldots, t)$$

where x *ranges over a given set of variables* $\mathcal{V}$*,* c *ranges over nullary function symbols in* $\mathcal{F}$*, and* f *ranges over function symbols in* $\mathcal{F}$ *with arity* $n > 0$*.*

Example 5. *If* n *is a nullary function symbol (constant),* f *is a unary function symbol, and* g *is a binary function symbol, then examples of terms are:*

- $g(f(n), n)$
- $f(g(n, f(n)))$

Example 6. *If* $0, 1, 2, 3$ *are nullary functions (constants),* s *is unary, and* $+, -$ *and* $*$ *are binary, then*

$$*(-(2, +(s(x), y)), x)$$

is a term.

Occasionally, we allow ourselves to use infix notation for function symbols as in

$$(2 - (s(x) + y)) * x$$

Definition 2. *The set of* formulas *in predicate logic is defined by the BNF:*

$$\begin{aligned}\phi \quad ::= \quad & P(t_1, t_2, \ldots, t_n) \mid \bot \mid \top \mid (\neg\phi) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid \\ & (\phi \to \phi) \mid (\forall x \phi) \mid (\exists x \phi)\end{aligned}$$

where $P \in \mathcal{P}$ is a predicate symbol of arity $n \geq 0$, t_i are terms over $\mathcal{F}$ and x is a variable.

We allow for nullary predicate symbols. The predicates that they denote do not depend on any arguments, and as such are similar to propositional atoms in propositional logic.

Convention 1. *Just like for propositional logic, we introduce convenient conventions to reduce the number of parentheses:*

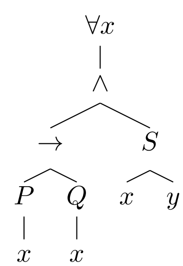
- $\neg, \forall x$ *and* $\exists x$ *bind most tightly;*
- *then* $\wedge$ *and* $\vee$*;*
- *then* $\to$*, which is right-associative.*

We extend the the notion of a *parse tree*, to provide for functions, predicates and quantifiers.

Example 7.

$$\forall x((P(x) \to Q(x)) \wedge S(x, y))$$

has parse tree



2.1 Equality

Equality is a common predicate, usually used in infix notation.

$$= \in \mathcal{P}$$

Example 8. *Instead of the formula*

$$= (f(x), g(x))$$

we usually write the formula

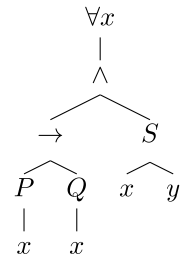
$$f(x) = g(x)$$

2.2 Free and Bound Variables

Consider the formula

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

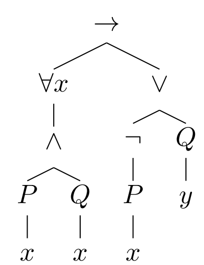
with the following syntax tree:



The quantifier $\forall x$ refers to all *occurrences* of x below it in the syntax tree. We say that the quantifier *binds* the variable occurrence. The variable occurrence x is said to be *bound* by $\forall x$. A variable that is not bound by any quantifier is called *free*. For example, the variable y is a free variable in the formula above. Consider the formula

$$(\forall x(P(x) \wedge Q(x))) \rightarrow (\neg P(x) \vee Q(y))$$

with the following syntax tree:



Here, the occurrences of x in $P(x) \wedge P(x)$ are bound by $\forall x$, whereas the occurrence of x in $\neg P(x)$ is free.

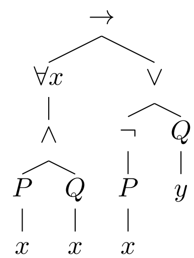
In order to define the semantics of quantifiers, we need to be able to replace free occurrences of variables systematically by terms, using an operation called *substitution*.

Definition 3. *Given a variable x, a term t and a formula ϕ, we define $[x \Rightarrow t]\phi$ to be the formula obtained by replacing each free occurrence of variable x in ϕ with t.*

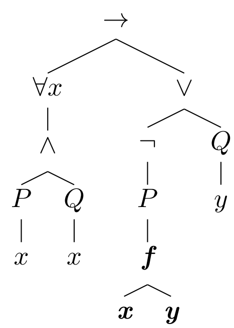
Example 9.

$$[x \Rightarrow f(x, y)]((\forall x(P(x) \wedge Q(x))) \rightarrow (\neg P(x) \vee Q(y)))$$

$$= \forall x(P(x) \wedge Q(x))) \rightarrow (\neg P(f(x, y)) \vee Q(y))$$



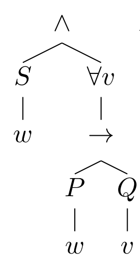
becomes



The notion of substitution of x by t in ϕ, denoted $[x \Rightarrow t]\phi$ poses a technical difficulty when t contains a variable y and x occurs under the scope of $\forall y$ in ϕ.

Example 10.

$$[w \Rightarrow f(v, v)](S(w) \wedge \forall v(P(w) \rightarrow Q(v)))$$



Here the variable v occurs in the term that is to be substituted for w. However, there is an occurrence of w under a $\forall w$. Thus, a naive execution of the substitution would "slip" occurrences of w "under" the scope of $\forall w$. This is to be avoided; any variable in t needs to be free in $[x \Rightarrow t]\phi$.

Definition 4. *Given a term t, a variable x and a formula ϕ, we say that t is* free for x in ϕ, *if no free x leaf in ϕ occurs in the scope of $\forall y$ or $\exists y$ for any variable y occurring in t.*

In order to compute $[x \Rightarrow t]\phi$, we demand that t is free for x in ϕ. If this condition does not hold, we consistently rename bound variables in ϕ.

Example 11.

$$[w \Rightarrow f(v, v)](S(w) \wedge \forall v(P(w) \rightarrow Q(v)))$$

$$\Downarrow$$

$$[w \Rightarrow f(v,v)](S(w) \wedge \forall z(P(w) \to Q(z)))$$

$$\Downarrow$$

$$S(f(v,v)) \wedge \forall z(P(f(v,v)) \to Q(z))$$

3 Semantics

3.1 Models

Definition 5. *Let $\mathcal{F}$ contain function symbols and $\mathcal{P}$ contain predicate symbols. A model $\mathcal{M}$ for $(\mathcal{F}, \mathcal{P})$ consists of:*

1. *A non-empty set U, the* universe*;*
2. *for each nullary function symbol $f \in \mathcal{F}$ a concrete element $f^{\mathcal{M}} \in U$;*
3. *for each $f \in F$ with arity $n > 0$, a concrete function $f^{\mathcal{M}} : U^n \to U$;*
4. *for each $P \in \mathcal{P}$ with arity $n > 0$, a function $P^{\mathcal{M}} : U^n \to \{F, T\}$.*
5. *for each $P \in \mathcal{P}$ with arity $n = 0$, a value from $\{F, T\}$.*

Example 12. *Let $\mathcal{F} = \{e, \cdot\}$ and $\mathcal{P} = \{\leq\}$.*
Let model $\mathcal{M}$ for $(\mathcal{F}, \mathcal{P})$ be defined as follows:

1. *Let U be the set of binary strings over the alphabet $\{0, 1\}$;*
2. *let $e^{\mathcal{M}} = \epsilon$, the empty string;*
3. *let $\cdot^{\mathcal{M}}$ be defined such that $s_1 \cdot^{\mathcal{M}} s_2$ is the concatenation of the strings s_1 and s_2; and*
4. *let $\leq^{\mathcal{M}}$ be defined such that $s_1 \leq^{\mathcal{M}} s_2$ iff s_1 is a prefix of s_2.*

Examples of elements of U are ϵ and 10001. The term $1010 \cdot 1100$ is given the meaning $1010 \cdot^{\mathcal{M}} 1100 = 10101100$ in $\mathcal{M}$, whereas the term $000 \cdot \epsilon$ is given the meaning $000 \cdot^{\mathcal{M}} \epsilon = 000$.

3.2 Equality Revisited

Usually, we require that the equality predicate $=$ is interpreted as same-ness. This means that allowable models are restricted to those in which $a =^{\mathcal{M}} b$ holds if and only if a and b are the same elements of the model's universe.

Example 13. *Continuing Example 12, we require in every model* $\mathcal{M}$ *that* $000 =^{\mathcal{M}} 000$ *holds and that* $001 =^{\mathcal{M}} 100$ *does not hold. We write* $001 \neq^{\mathcal{M}} 100$ *to denote the latter.*

Example 14. *Let* $\mathcal{F} = \{z, s\}$ *and* $\mathcal{P} = \{\leq\}$.
Let model $\mathcal{M}$ *for* $(\mathcal{F}, \mathcal{P})$ *be defined as follows:*

1. *Let* U *be the set of natural numbers;*
2. *let* $z^{\mathcal{M}} = 0$;
3. *let* $s^{\mathcal{M}}$ *be defined such that* $s(n) = n + 1$; *and*
4. *let* $\leq^{\mathcal{M}}$ *be defined such that* $n_1 \leq^{\mathcal{M}} n_2$ *iff the natural number* n_1 *is less than or equal to* n_2.

With the above restriction on equality, we can see that the relation $=^{\mathcal{M}}$ *is a subset of* $\leq^{\mathcal{M}}$; *we write* $=^{\mathcal{M}} \subseteq \leq^{\mathcal{M}}$.

3.3 Free Variables and the Satisfaction Relation

We can give meaning to formulas with free variables by providing an environment (lookup table) that assigns variables to elements of our universe:

$$l : \mathcal{V} \to U.$$

We define environment extension such that $l[x \mapsto a]$ is the environment that maps x to a and any other variable y to $l(y)$. Using this definition, we can now define when a model satisfies a formula.

Definition 6. *The model* $\mathcal{M}$ *satisfies* ϕ *with respect to environment* l*, written* $\mathcal{M} \models_l \phi$:

- *in case* ϕ *is of the form* $P(t_1, t_2, \ldots, t_n)$*, if* $a_1, a_2, \ldots, a_n$ *are the results of evaluating* $t_1, t_2, \ldots, t_n$ *with respect to* l*, and if* $P^{\mathcal{M}}(a_1, a_2, \ldots, a_n) = T$;
- *in case* ϕ *is of the form* P*, if* $P^{\mathcal{M}} = T$;
- *in case* ϕ *has the form* $\forall x \psi$*, if the* $\mathcal{M} \models_{l[x \mapsto a]} \psi$ *holds for all* $a \in U$;
- *in case* ϕ *has the form* $\exists x \psi$*, if the* $\mathcal{M} \models_{l[x \mapsto a]} \psi$ *holds for some* $a \in U$;
- *in case* ϕ *has the form* $\neg \psi$*, if* $\mathcal{M} \models_l \psi$ *does not hold;*
- *in case* ϕ *has the form* $\psi_1 \vee \psi_2$*, if* $\mathcal{M} \models_l \psi_1$ *holds or* $\mathcal{M} \models_l \psi_2$ *holds;*

- *in case ϕ has the form $\psi_1 \wedge \psi_2$, if $\mathcal{M} \models_l \psi_1$ holds and $\mathcal{M} \models_l \psi_2$ holds; and*
- *in case ϕ has the form $\psi_1 \to \psi_2$, if $\mathcal{M} \models_l \psi_2$ holds whenever $\mathcal{M} \models_l \psi_1$ holds.*

If a formula ϕ has no free variables, we call ϕ a *sentence*. In this case, $\mathcal{M} \models_l \phi$ holds or does not hold regardless of the choice of l. Thus for sentences ϕ, we leave out the environment, and write $\mathcal{M} \models \phi$ or $\mathcal{M} \not\models \phi$.

Definition 7. *Let Γ be a possibly infinite set of formulas in predicate logic and ψ a formula. We say that Γ* entails *ψ, written $\Gamma \models \psi$, iff for all models $\mathcal{M}$ and environments l, whenever $\mathcal{M} \models_l \phi$ holds for all $\phi \in \Gamma$, then $\mathcal{M} \models_l \psi$.*

Definition 8. *We say that a formula ψ is* satisfiable*, iff there is some model $\mathcal{M}$ and some environment l such that $\mathcal{M} \models_l \psi$ holds.*

Definition 9. *A set of formulas Γ is called* satisfiable*, iff there is some model $\mathcal{M}$ and some environment l such that $\mathcal{M} \models_l \phi$, for all $\phi \in \Gamma$.*

Definition 10. *Let Γ be a possibly infinite set of formulas in predicate logic and ψ a formula. The formula ψ is called* valid*, iff for all models $\mathcal{M}$ and environments l, we have $\mathcal{M} \models_l \psi$.*

Note that both validity and entailment require to consider all possible models. Not only are we free to choose the universe, we are also free to decide the interpretation of every function and predicate symbol. As a result, the number of models is usually infinite (and usually not countably infinite). This makes it very hard to prove validity and entailment, using semantic techniques.

The question arises how to effectively argue about all possible models. Would it be possible to define a method of natural deduction that allows us to answer the questions of entailment and validity?

4 Proof Theory

4.1 Inheriting Natural Deduction of Propositional Logic

If we consider propositions as nullary predicates, propositional logic is a sublanguage of predicate logic. It will come as no surprise that we can translate the rules for natural deduction in propositional logic directly to predicate logic. Each of the following rules applies to any formulas ϕ and ψ of predicate logic.

$$\frac{\phi \qquad \psi}{\phi \wedge \psi}[\wedge i] \qquad \frac{\phi \wedge \psi}{\phi}[\wedge e_1] \qquad \frac{\phi \wedge \psi}{\psi}[\wedge e_2]$$

$$\frac{\phi}{\phi \vee \psi}\,[\vee i_1] \qquad \frac{\psi}{\phi \vee \psi}\,[\vee i_2] \qquad \frac{\phi \vee \psi \quad \boxed{\begin{array}{c}\phi\\ \vdots\\ \chi\end{array}} \quad \boxed{\begin{array}{c}\psi\\ \vdots\\ \chi\end{array}}}{\chi}\,[\vee e]$$

$$\frac{\boxed{\begin{array}{c}\phi\\ \vdots\\ \psi\end{array}}}{\phi \to \psi}\,[\to i] \qquad \frac{\phi \qquad \phi \to \psi}{\psi}\,[\to e]$$

$$\frac{\boxed{\begin{array}{c}\phi\\ \vdots\\ \bot\end{array}}}{\neg\phi}\,[\neg i] \qquad \frac{\phi \qquad \neg\phi}{\bot}\,[\neg e]$$

$$\frac{\bot}{\phi}\,[\bot e] \qquad \frac{\neg\neg\phi}{\phi}\,[\neg\neg e]$$

4.2 Equality

We mentioned in "Semantics of Predicate Logic" that equality is usually interpreted to mean identity, which means that in a model $a =^{\mathcal{M}} b$ holds if and only if a and b are the same elements of the model's universe. It is safe to assume $t = t$ for any term t, because both sides of the equation will evaluate to the same element, regardless of the context (environment) in which we operate. The following equality introduction rule expresses this reasoning.

$$\frac{}{t = t}[= i]$$

The next rule, equality elimination, allows us to replace a term t_1 by another term t_2, provided that $t_1 = t_2$ is already proven. More precisely, in order to prove a formula ψ, in which a term t_2 appears (possibly multiple times), it is sufficient to prove $t_1 = t_2$ and the formula ψ' that results from ψ by replacing t_2 by t_1. The rule stated below uses a formula ϕ in which a free variable x represents the positions of t_2 in ψ, thus $\psi = [x \Rightarrow t_2]\phi$, and $\psi' = [x \Rightarrow t_1]\phi$.

$$\frac{t_1 = t_2 \qquad [x \Rightarrow t_1]\phi}{[x \Rightarrow t_2]\phi}[= e]$$

Using these two rules, we show: We show:

$$f(z) = g(z) \vdash h(g(z)) = h(f(z))$$

as follows:

1	$f(z) = g(z)$	premise
2	$h(f(z)) = h(f(z))$	$= i$
3	$h(g(z)) = h(f(z))$	$= e$ 1,2

Note that the formula $h(g(z)) = h(f(z))$ in Line 3 has the form $[x \Rightarrow t_2]\phi$, where t_2 is $g(z)$ and ϕ is $h(z) = h(f(z)$. If we use $f(z)$ for t_1, then Rule $= e$ asks us to prove $t_1 = t_2$ (Line 1), and $[x \Rightarrow t_1]\phi$ (Line 2).

4.3 Universal Quantification

Elimination of Universal Quantification Once you have proven $\forall x\phi$, you can replace x by any term t in ϕ, provided that t is free for x in ϕ, and thus "eliminate" the universal quantification.

$$\frac{\forall x\phi}{[x \Rightarrow t]\phi}[\forall x\ e]$$

This rule is justified by the semantics of $\forall x\phi$, since in a particular context (environment) any term t denotes a value in the model, and ϕ holds for all such values, if $\forall x\phi$ holds in the model.

In t any function symbols of the logic, as well as variables that are known in the context can be used.

Example 15. *We shall prove:* $S(g(john)), \forall x(S(x) \rightarrow \neg L(x)) \vdash \neg L(g(john))$

1	$S(g(john))$	*premise*
2	$\forall x(S(x) \rightarrow \neg L(x))$	*premise*
3	$S(g(john)) \rightarrow \neg L(g(john))$	$\forall x$ *e 2*
4	$\neg L(g(john))$	$\rightarrow$ *e 3,1*

Introduction of Universal Quantification The introduction rule for universal quantification is more complicated. Let us consider a new kind of box that allows us to introduce a fresh variable. For example,

$$\boxed{\vdots}^{\,z}$$

is a box in which the variable z can be used in terms, as in

$$\boxed{\begin{array}{c}\vdots\\ f(z) = f(z)\\ \vdots\end{array}}^{\,z}$$

Let us say we introduce a variable x_0 in a box. Without any assumptions on x_0, we prove a formula ψ, in which x_0 appears. The fact that x_0 appears in ψ, we can characterize by writing ψ as $[x \Rightarrow x_0]\phi$. Since we have not made any assumptions on x_0 within the box, we have shown that $[x \Rightarrow x_0]\phi$ holds for all possible instantiations of x by values of the universe; in other words, we can conclude $\forall x\phi$.

$$\frac{\boxed{\begin{array}{c}\vdots\\ [x \Rightarrow x_0]\phi\end{array}}^{\,x_0}}{\forall x\phi}\,[\forall x\ i]$$

The variable x_0 must be *fresh*; we cannot introduce the same variable twice in nested boxes. Freshness of course guarantees that x_0 is free for x in ϕ.

Example 16. *We shall prove the sequent* $\forall x(P(x) \rightarrow Q(x)), \forall x P(x) \vdash \forall x Q(x)$

1	$\forall x(P(x) \rightarrow Q(x))$	*premise*	
2	$\forall x P(x)$	*premise*	
3	$P(x_0) \rightarrow Q(x_0)$	$\forall x$ *e 1*	x_0
4	$P(x_0)$	$\forall x$ *e 2*	
5	$Q(x_0)$	$\rightarrow$ *e 3,4*	
6	$\forall x Q(x)$	$\forall x$ *i 3–5*	

4.4 Existential Quantification

Introduction of Existential Quantification For existential quantification, the easy direction is introduction.

$$\frac{[x \Rightarrow t]\phi}{\exists x \phi}[\exists x\ i]$$

In order to prove $\exists x \phi$, it suffices to find a term t as "witness", provided—as usual—that t is free for x in ϕ.

Example 17. *Assume that the set $\mathcal{F}$ contains a nullary function symbol c, and that the set $\mathcal{P}$ contains a unary predicate symbol P. We should be able to prove:*

$$\forall x P(x) \vdash \exists x P(x)$$

since at least the constant c should have the property P, once we know that all elements of the universe has the property P. The corresponding proof follows:

1	$\forall x P(x)$	*premise*
2	$[x \Rightarrow c]P(x)$	$\forall x$ *e 1*
3	$\exists x P(x)$	$\exists x$ *i 2*

Elimination of Existential Quantification Finally, for elimination of existential quantification, we combine the two kinds of boxes; we simultaneously introduce a fresh variable *and* an assumption.

$$\frac{\exists x \phi \qquad \boxed{\begin{array}{c}[x \Rightarrow x_0]\phi \\ \vdots \\ \chi\end{array}}^{x_0}_{[x \Rightarrow x_0]\phi}}{\chi}[\exists e]$$

If we know $\exists x \phi$, we know that there exist at least one object x for which ϕ holds. We call that element x_0, and assume $[x \Rightarrow x_0]\phi$ within a box in which we introdue x_0. Without assumptions on x_0, we prove a formula χ, in which x_0 does not appear. Since we have not made any assumptions on x_0, we can conclude from $\exists x \phi$ that χ holds.

Example 18. *We prove the following sequent:*

$$\forall x (P(x) \to Q(x)), \exists x P(x) \vdash \exists x Q(x)$$

1	$\forall x(P(x) \to Q(x))$	*premise*	
2	$\exists x P(x)$	*premise*	
3	$P(x_0)$	*assumption*	x_0
4	$P(x_0) \to Q(x_0)$	$\forall x$ *e 1*	
5	$Q(x_0)$	$\to$ *e 4,3*	
6	$\exists x Q(x)$	$\exists x$ *i 5*	
7	$\exists x Q(x)$	$\exists x$ *e 2,3–6*	

Note that $\exists x Q(x)$ *within the box does not contain* x_0*, and therefore can be "exported" from the box.*

Example 19. *We prove the following sequent:*

$$\forall x(Q(x) \to R(x)), \exists x(P(x) \wedge Q(x)) \vdash \exists x(P(x) \wedge R(x))$$

1	$\forall x(Q(x) \to R(x))$	*premise*	
2	$\exists x(P(x) \wedge Q(x))$	*premise*	
3	$P(x_0) \wedge Q(x_0)$	*assumption*	x_0
4	$Q(x_0) \to R(x_0)$	$\forall x$ *e 1*	
5	$Q(x_0)$	$\wedge e_2$ *3*	
6	$R(x_0)$	$\to$ *e 4,5*	
7	$P(x_0)$	$\wedge e_1$ *3*	
8	$P(x_0) \wedge R(x_0)$	$\wedge i$ *7, 6*	
9	$\exists x(P(x) \wedge R(x)$	$\exists x$ *i 8*	
10	$\exists x(P(x) \wedge R(x))$	$\exists x$ *e 2,3–9*	

Note that variables introduced by a box must be fresh! The following is not a proof, since the variable x_0 is introduced in nested boxes.

1	$\exists x P(x)$	premise	
2	$\forall x(P(x) \to Q(x))$	premise	
3			x_0
4	$P(x_0)$	assumption	x_0
5	$P(x_0) \to Q(x_0)$	$\forall x$ e 2	
6	$Q(x_0)$	$\to e$ 5,4	
7	$Q(x_0)$	$\exists x$ e 1, 4–6	
8	$\forall y Q(y)$	$\forall y$ i 3–7	

4.5 Equivalences

We write $\phi \dashv\vdash \psi$ iff $\phi \vdash \psi$ and also $\psi \vdash \phi$.

Lemma 1.

$$
\begin{aligned}
\neg\forall x\phi &\dashv\vdash \exists x\neg\phi \\
\neg\exists x\phi &\dashv\vdash \forall x\neg\phi \\
\forall x\forall y\phi &\dashv\vdash \forall y\forall x\phi \\
\exists x\exists y\phi &\dashv\vdash \exists y\exists x\phi \\
\forall x\phi \land \forall x\psi &\dashv\vdash \forall x(\phi \land \psi) \\
\exists x\phi \lor \exists x\psi &\dashv\vdash \exists x(\phi \lor \psi)
\end{aligned}
$$

Proof. We shall prove the left-to-right directions of the first and fourth statement, and leave the remaining proofs to the reader. The proves are actually schemas; actual sequents and proofs are obtained by replacing ϕ and ψ with arbitrary formulas in a particular predicate logic.

- $\neg\forall x\phi \vdash \exists x\neg\phi$

1	$\neg\forall x\phi$	premise	
2	$\neg\exists x\neg\phi$	assumption	
3			x_0
4	$\neg[x \Rightarrow x_0]\phi$	assumption	
5	$\exists x\neg\phi$	$\exists x\ i$ 4	
6	$\bot$	$\neg e$ 5, 2	
7	$[x \Rightarrow x_0]\phi$	PBC 4–6	
8	$\forall x\phi$	$\forall x\ i$ 3–7	
9	$\bot$	$\neg e$ 8, 1	
10	$\exists x\neg\phi$	PBC 2–9	

- $\exists x\exists y\phi \vdash \exists y\exists x\phi$

  If x and y are the same variable, the left and write hand side are the same formula, and thus the sequent holds through a simple argument (for example conjunction introduction followed by elimination).

  Assume now that x and y are different variables.

1	$\exists x\exists y\phi$	premise	
2	$[x \Rightarrow x_0](\exists y\phi)$	assumption	x_0
3	$\exists y([x \Rightarrow x_0]\phi)$	def of subst (x, y different)	
4	$[y \Rightarrow y_0][x \Rightarrow x_0]\phi$	assumption	y_0
5	$[x \Rightarrow x_0][y \Rightarrow y_0]\phi$	def of subst (x, y, x_0, y_0 different)	
6	$\exists x[y \rightarrow y_0]\phi$	$\exists x\ i$ 5	
7	$\exists y\exists x\phi$	$\exists y\ i$ 6	
8	$\exists y\exists x\phi$	$\exists y\ e$ 3, 4–7	
9	$\exists y\exists x\phi$	$\exists x\ e$ 1, 2–8	

□

Exercise 1. *Prove the remaining directions of the statements in Lemma 1.*

Lemma 2. *Assuming that x is not free in ψ, the following sequents hold:*

$$\begin{array}{rcl} \forall x\phi \wedge \psi & \dashv\vdash & \forall x(\phi \wedge \psi) \\ \forall x\phi \vee \psi & \dashv\vdash & \forall x(\phi \vee \psi) \\ \exists x\phi \wedge \psi & \dashv\vdash & \exists x(\phi \wedge \psi) \\ \exists x\phi \vee \psi & \dashv\vdash & \exists x(\phi \vee \psi) \end{array}$$

Exercise 2. *Prove the statements of Lemma 2.*

5 Soundness, Completeness, Undecidability

The following result justifies the use of natureal deduction in predicate logic.

Theorem 1 (Soundness and Completeness of Predicate Logic)**.**

$$\phi_1, \ldots, \phi_n \models \psi$$

iff

$$\phi_1, \ldots, \phi_n \vdash \psi$$

The theorem states that every valid sequent can be proven, and every sequent that can be proven is valid. This theorem was proven by Kurt Gödel in 1929 in his doctoral dissertation. A description of his proof, as well as the proofs of the following theorems, is beyond the scope of this chapter.

Theorem 2. *The decision problem of validity in predicate logic is undecidable: no program exists which, given any language in predicate logic and any formula ϕ in that language, decides whether $\models \phi$.*

Proof. (sketch)

- Establish that the Post Correspondence Problem (PCP) is undecidable
- Translate an arbitrary PCP, say C, to a formula ϕ.
- Establish that $\models \phi$ holds if and only if C has a solution.
- Conclude that validity of predicate logic formulas is undecidable.

□