Module 13
Non-Modal Predicate Logic

G. J. Mattey

May 31, 2007

Contents

1 The Syntax of Predicate Logic

Predicate Logic with identity (*PL*) is language which is an extension of Sentence Logic (*SL*), adding to it some further vocabulary and formation rules.[1] The treatment here follows that of Richmond Thomason in *Symbolic Logic: An Introduction* (1970), which deviates in some details from better-known presentations of Predicate Logic. The reason is that Thomason's system is more readily adaptable to modal predicate logic than are the standard systems.

1.1 The Expressions of Predicate Logic

The language *PL* is an extension of the language *SL*. Sentence letters of *SL*, *falsum*, the *SL* operators, and the *SL* punctuation marks are carried over from *SL* to *PL*. Predicate Logic adds several new kinds of expression to this base. We begin with a set of *terms*, of which there are two kinds: *constants* and *parameters*.[2] Constants are lower-case Roman letters from 'a' to 't,' with or without positive integer subscripts. Parameters are the lower case letters 'u' and 'v,' with or without positive integer subscripts. There is also a set of variables, which are lower-case letters from 'w' to 'z,' again with or without positive integer subscripts.[3]

Next, there is a set of *predicate letters*, which are capital Roman letters with a positive integer superscript, such as 'F^2'. The superscript indicates the number of terms, variables, or mixtures of terms and variables, that have to follow the predicate letter to create a formula of *PL*.[4] The predicate letters may also carry a positive integer subscript, so 'F^2_3' is a predicate letter.

A special predicate is the two-place *identity* predicate. As it will be given its own semantical rule and rules of inference, we will use a special symbol, '=,' to represent identity.

Finally, there are two *quantifier symbols*, '∃' and '∀,' called the *existential* and *universal* quantifier symbols, respectively. A pair consisting of a quantifier symbol and a variable, enclosed in parentheses, is a *quantifier*. Thus '$(\forall x)$' and '$(\exists y)$' are quantifiers.[5]

We summarize the set of expressions of **PL** as follows:

Expressions of *PL*

- An infinitely large set of *sentence letters*: $A, B, C, \ldots, Z, A_1, B_1, \ldots, Z_1, A_2, B_2, \ldots$.
- A *sentential constant*: '⊥.'
- Two *punctuation marks*: '(' and ').'
- A set of five *operators*: '∼', '∧', '∨', '⊃', and '≡.'
- A set of *constants*: $a, b, \ldots, t, a_1, b_1, \ldots, t_1, a_2, b_2, \ldots$.
- A set of *parameters*: $u, u_1, u_2, \ldots, v, v_1, v_2, \ldots$.
- A set of *variables*: $w, x, y, z, w_1, x_1, y_1, z_1, w_2, x_2, \ldots$.

[1] For simplicity, we depart from the nomenclature of *The Logic Book*, which distinguishes between Predicate Logic proper and Predicate Logic with Identity, which they call *PI*. We will be dealing exclusively with Predicate Logic with Identity here.

[2] Constants are called "names" in some texts. What are here called simply "parameters" are called "individual parameters" by Thomason.

[3] In most texts, variables are classified as terms along with constants. Here, the standard role of variables as terms is taken over by parameters.

[4] The superscript may be suppressed for readability when it is followed by *n* terms.

[5] It is common for the quantifier symbols to be referred to as "quantifiers," but strictly speaking this is incorrect.

- A set of *predicate letters*: $A^1, \ldots, Z^1, A^1_1, \ldots, Z^1_1, A^1_2, \ldots A^2, \ldots, Z^2, A^2_1, \ldots, Z^2_1, A^2_2, \ldots$.
- A special two-place predicate symbol '$=$.'
- A set of two *quantifier symbols*: '$\forall$,' '$\exists$.'

1.2 Rules of Formation for *PL*

To generate the rules of formation for *PL*, we must add to our stock of meta-variables. As with the operators in *SL*, we shall take quantifier symbols and the identity predicate to be names of themselves. For the other new expressions of *PL*, we will use bold-face symbols, with or without subscripts. For constants, we will use '**a**,' for variables, '**x**,' '**y**,' and '**z**' for parameters, '**u**' and '**v**' and for terms (i.e., constants or parameters) '**t**.' In each case, we will use no subscript or italicized roman lower-case subscripts for unspecified constants, variables, parameters and more generally terms, and positive integer subscripts for specific constants, variables, parameters, and terms. For 1-place predicates, we will use '$\mathbf{P}^1$,' with or without a positive integer subscript, and so-on for predicates of more than one place. We will also refer more generally to *n*-place predicates, writing '$\mathbf{P}^n$.'

An *atomic sentence* of *PL* will be either a sentence letter, *falsum*, or an *n*-place predicate letter followed by *n* terms. In Sentence Logic, non-atomic, or *compound* sentences can be built from atomic sentences through the use of *SL*-operators. The more interesting compound sentences are built using quantifiers.

It will take some preliminary work to define a sentence that is built from a simpler sentence by prefixing a quantifier. We begin by letting 'Φ' stand for any string of expressions of *PL*. We will also use the meta-variables **b** and **c** to stand for constants, parameters, or variables. When a string Φ contains an occurrence of **b**, then we will write '$\Phi(\mathbf{b})$.' Then we can symbolize the result of substituting **c** for all occurrences of **b** within a the string Φ. We will use the notation $\Phi(\mathbf{c}/\mathbf{b})$ to symbolize such a substitution. In every string of this kind, a substitution must have been made, and so **b** was in the original string of expressions.

For example, if the string is '*xxxxxx*,' then the result of substituting '*u*' for all occurrences of '*x*' is '*uuuuuu*.' If Φ is '$\sim F^2au$' (which will turn out to be a sentence of *PL*), then '$\sim F^2ax$.' is the result of substituting '*x*' for '*u*' in the sentence. In the case where Φ is a sentence α of *PL*, we write '$\alpha(b)$' and '$\alpha(\mathbf{c}/\mathbf{b})$.'

Before stating the formation rules, we must provide some definitions which will be used in some of them. These definitions are stated hypothetically, leaving undefined what a sentence of *PL* might be. Assume that $\alpha(\mathbf{u})$ is a sentence of *PL*. Then a *quantified sentence* of *PL* is the result of replacing all the occurrences of **u** with occurrences of **x**, and then affixing a quantifier in front of the sentence. There are two kinds of quantified sentences: $(\forall\mathbf{x})\alpha(\mathbf{x}/\mathbf{u})$ and $(\exists\mathbf{x})\alpha(\mathbf{x}/\mathbf{u})$.

A quantified sentence is always formed from a sentence which does not contain the variable in the quantifier, but contains instead a parameter for which the variable is substituted. We will refer to the result of making the substitution (but before the affixing of the quantifier) a *quasi-sentence* of *PL*.[6]

We further define the *scope* of the quantifier to be the quasi-sentence following the quantifier, which sentence is said to be *governed by* the quantifier, Any variable **x** in the quasi-sentence used to build the quantified sentences $(\forall\mathbf{x})\alpha(\mathbf{x}/\mathbf{u})$ or $(\exists\mathbf{x})\alpha(\mathbf{x}/\mathbf{u})$ is said to be *bound* by the respective quantifier.

A parameter **u** occurring in a sentence $\alpha(\mathbf{u})$ is said to be *free for* a variable **v** if and only if **u** does not occur in the scope of a quantifier containing the variable **x**. Finally, we will say that $(\forall\mathbf{x})\alpha(\mathbf{x}/\mathbf{u})$ is a *universal quantification* of a sentence $\alpha(\mathbf{u})$, where **u** is free for **x**. Similarly, $(\exists\mathbf{x})\alpha(\mathbf{x}/\mathbf{u})$ is an *existential quantification* of a sentence $\alpha(\mathbf{u})$, where **u** is free for **x**.

[6] In the literature, such a string of expressions is often called an "open sentence."

We will give examples of the use of these definitions after stating the formation rules, which will define the sentences we use in the examples.

Formation Rules of *PL*

1. All sentence letters of *SL* are sentences of *PL*.
2. '$\bot$' is a sentence of *PL*.
3. If $\mathbf{P}^n$ is a predicate letter of *PL* and $\mathbf{t_1}, \ldots, \mathbf{t_n}$ are terms of *PL*, then $\mathbf{P}^n\mathbf{t}_1, \ldots, \mathbf{t}_n$ is a sentence of *PL*
4. If $\mathbf{t}_1$ and $\mathbf{t}_2$ are terms of *PL*, then $\mathbf{t}_1 = \mathbf{t}_2$ is a sentence of *PL*.
5. If α is a sentence of *PL*, then $\sim\alpha$ is a sentence of *PL*.
6. If α and β are sentences of *PL*, then $(\alpha \wedge \beta)$ is a sentence of *PL*.
7. If α and β are sentences of *PL*, then $(\alpha \vee \beta)$ is a sentence of *PL*.
8. If α and β are sentences of *PL*, then $(\alpha \supset \beta)$ is a sentence of *PL*.
9. If α and β are sentences of *PL*, then $(\alpha \equiv \beta)$ is a sentence of *PL*.
10. If $\alpha(\mathbf{u})$ is a sentence of *PL* and $\mathbf{u}$ is free for $\mathbf{x}$ in $\alpha(\mathbf{u})$, then $(\forall\mathbf{x})\alpha(\mathbf{x}/\mathbf{u})$ is a sentence of *PL*.
11. If $\alpha(\mathbf{u})$ is a sentence of *PL* and $\mathbf{u}$ is free for $\mathbf{x}$ in $\alpha(\mathbf{u})$, then $(\exists\mathbf{x})\alpha(\mathbf{x}/\mathbf{u})$ is a sentence of *PL*.
12. Nothing else is a sentence of *PL*.

Having stated the formation rules defining a sentence of *PL*, we may now illustrate their use with some examples. Because 'F^2' is a two-place predicate letter of *PL* and 'u' and 'v' are parameters of *PL*, 'F^2vu' is a sentence of *PL* by clause 3. Since the sentence contains no quantifiers, it is not in the scope of any quantifiers. Thus the parameters 'u' and 'v' are free for all variables. Because of this, we can form a quantified sentence '$(\forall x)F^2vx$.' This is done by first generating the quasi-sentence 'F^2vx' and then affixing the quantifier to its front. The quasi-sentence is then in the scope of, and is governed by, the quantifier '$(\forall x)$.' The variable 'x' occurring in '$(\forall x)F^2vx$' is bound to its occurrence in '$(\forall x)$.' In the newly generated sentence, we have a remaining parameter, 'v,' which is in the scope of a quantifier containing 'x.' Therefore, 'v' is not free for 'x' in '$(\forall x)F^2vx$.'

If we were to add a conjunct to '$(\forall x)F^2vx$,' producing the conjunction '$(\forall x)F^2vx \wedge (\exists x)G^2ax$,' the result would not be a quantified sentence, and the occurrence of 'x' in the second conjunct would be bound only by the quantifier '$(\exists x)$' and not by the quantifier '$(\forall x)$.'

Now we will note two important consequences of clauses 10 and 11, which guarantee that quantifiers can be prefixed only to sentences which originally contain a parameter. The first consequence is that there can be no "vacuous" quantification, in which the variable in the quantifier is not followed by a string of expressions containing the variable, as for example in the non-sentence '$(\forall x)\sim F^2vu$.' The definition of $\alpha(\mathbf{x}/\mathbf{u}$ stipulates that at least one parameter must be replaced by a variable contained in the quantifier.

The second consequence is that there can be no "redundant" quantification, in which the variable in the quantifier does occur in the following string but occurs there only because it was already substituted for a parameter by clause 10 or 11. This is the point of the requirement that the parameter be free for the variable. In our example, the parameter 'v' in the sentence '$(\forall x)\sim F^2vx$' is not free for 'x.' An attempt to form a universal quantification of this sentence using 'x' once again, i.e., '$(\forall x)(\forall x)\sim F^2xx$,' would for this reason be blocked.

2 Formal Semantics for *PL*

The semantics for *PL*-sentences is much more complicated than that for *SL*-sentences. This reflects the potentially greater complexity of *PL*-sentences.

2.1 The Domain of Discourse

To generate the semantical system *PI*, we expand the notion of an interpretation and its valuation function to interpret the additional syntactical expressions of *PL*. The interpretation of Sentence Letters and *falsum* remain as with *SI*. In addition, each *PI*-interpretation has as an element a *domain* or *universe* of *discourse*, which consists of a set of objects. It is required that the domain contain at least one item. The a *PI* interpretation is an ordered pair $\langle \mathbf{D}, \mathbf{v} \rangle$.

2.2 The Valuation Function

The valuation function **v** of an interpretation maps all the constants and parameters of *PL* into the domain. That is, each constant and each parameter is associated with one and only one member of the domain. (Note that a member of the domain may be designated by more than one constant or parameter by **v** in an interpretation.) So we say that $\mathbf{v_I(a)} \in \mathbf{D}$ and $\mathbf{v_I(u)} \in \mathbf{D}$. In non-modal Predicate Logic, parameters function semantically like constants, the difference between them showing up only in the derivational system. But in Modal Predicate Logic, there will in some systems be a substantial difference between the interpretation of constants and the interpretations of parameters.

Each predicate of *PL* is *n*-placed. The valuation function maps an *n*-place predicate $\mathbf{P}^n$ into the set $\mathbf{D}^n$ of all ordered *n*-tuples drawn from the domain.[7] $\mathbf{v_I(P^{\mathit{n}})} \in \mathbf{D}^n$. This set is known as the *extension* of the predicate. So, for example, for one interpretation I, a domain consists of the numbers 1 and 0, and the extension of a two-place predicate 'F^2' is $\{\langle 1, 1\rangle, \langle 0, 1\rangle\}$.

2.3 Variants of Valuation Functions

The interesting sentences of Predicate Logic are the ones which contain quantifiers. Universally quantified sentences say something about all members of the domain, and existentially quantified sentences say something about at least one member of the domain. To interpret quantified sentences, we need to work with the sentences from which they are formed, i.e., sentences containing parameters.

The key idea here is the notion of *variant* of a valuation function which assigns a member of the domain to a parameter **u**. We will first illustrate a variant and then define it. Suppose that our domain is D and that D $=\{1, 2\}$. Then we may have one interpretation I_1 where $\mathrm{v}_{\mathrm{I}_1}(u)$ = 1 and another interpretation I_2 where $\mathrm{v}_{\mathrm{I}_2}(u)$ = 2. Suppose further than on both interpretations the extension of 'F^1' is $\{\langle 1\rangle\}$. Now consider the sentence 'F^1u.' On I_1, 'F^1u' is true, and on I_2, 'F^1u' is false. We can say, however, that if $\mathrm{v}_{\mathrm{I}_2}$ were to be changed so that $\mathrm{v}_{\mathrm{I}_2}(\mathrm{u}) = 1$, then that variant valuation function would make 'F^1u' true.

More generally, we will let '**d**,' with or without integer subscripts, be a meta-variable standing for an item in the domain. In general, for some **d**, $\mathbf{v_I(u)} = \mathbf{d}$. In the previous example, we substituted a member of the domain d_1 (i.e., the number 1) for the member of the domain d_2 (i.e., the number 2) as the value of '*u*'. We write this as $\mathrm{v_I}[1/u]$, or more generally, $\mathbf{v}[\mathbf{d}_1/\mathbf{u}]$. The notation here is apt to be misleading, so it is worth describing carefully. We do not substitute **d** for **u**, but rather we substitute $\mathbf{d}_1$ for $\mathbf{d}_2$ as the *value* of '**u**.' So we should read a variant as being the result of substituting member of the domain $\mathbf{d}_1$, as the value

[7] This set is known as the *n*th Cartesian product of **D**.

of '$\mathbf{u}$,' for the member of the domain $\mathbf{d}_2$ that was the original value of '$\mathbf{u}$.' Thus produces a new valuation function, but one which has a specific relation to the initial function. The role of variants will become clear in the discussion of how quantified sentences get their truth-values.

2.4 Truth on an Interpretation

The semantical rules for *SI* carry over with no modification to *PI*. Here we will discuss only the peculiarities of interpeting sentences which are peculiar to *PL*.

An atomic sentence of *PL* is an n-place predicate followed by a string of n terms. Such a sentence is true on an interpretation just in case the ordered n-tuple consisting of the values of each of the terms is in the extension of the predicate. In the interpretation given above (where 'F^2' can be understood informally as standing for the relation of being less than or equal to), suppose 'a' is assigned by I the number 0, and 'b' as the number 1. Then 'F^2ab' will be a true sentence of *PL* on interpertation I because the ordered pair $\langle 1, 2\rangle$ is in the extension of 'F^2.'

In the special case of the identity predicate '=' flanked by terms, the sentence is true if and only if the terms on each side are assigned to the same the same individual by the valuation function. In the interpretation given in the last paragraph, if $\mathrm{v_I}(a) = 1$ and $\mathrm{v_I}(u) = 1$, then $\mathrm{v_I}(a = u) = \mathbf{T}$.[8]

A universally quantified sentence $(\forall\mathbf{x})(\alpha(\mathbf{x}/\mathbf{u}))$ is true on an interpretation $\mathbf{I}$ if and only if the sentence $\alpha(\mathbf{u})$ is true no matter what member of the domain is taken as the value of '$\mathbf{u}$.' That is, for all members $\mathbf{d}$ of the domain $\mathbf{D}$, $\mathbf{v}_{\mathrm{I}}[\mathbf{d/u}](\alpha(\mathbf{u})$ is true. That is to say, it does not matter which member of the domain the valuation function $\mathbf{v_I}$ might assign as the value of $\mathbf{u}$. All such assignments would make the sentence true. In our example above, the universally quantified sentence '$(\forall x)F^1x$' is false. The reason is that $\mathrm{v_I}[2/u](F^1u) = \mathbf{F}$.

An existentially quantified sentence is true on an interpretation if and only if there is at least one member of the domain of that interpretation such that, if it is taken as the value of $\mathbf{u}$, the sentence $\alpha(\mathbf{u})$ is true. In our example, we can see that $\mathrm{v_I}[1/u](F^1u) = \mathbf{T}$, and hence that $\mathrm{v_I}((\exists x)F^1x) = \mathbf{T}$.

There will be cases in which quantifiers occur in the scope of other quantifiers. For example, we have the sentence '$(\forall x)(\exists y)G^2xy$).' Suppose that for an interpretation I, the domain is the natural numbers and the relation that 'G^2' is interpreted as standing for is that of being greater than. So, informally, the sentence says that for each natural number, there is a greater natural number. The sentence is true on I just in case for all members $\mathbf{d}_i$ of the domain, $\mathrm{v_I}[\mathbf{d}_i/u]((\exists y)G^2uy) = \mathbf{T}$. But under what condition is this sentence true for all variants of v? Only when for each of them, there is a variant of it (which is a variant of a variant of v) which makes the core sentence G^2uv true. We indicate the variant of the variant with a comma between substitution conditions, where the variant is noted first and the variant of the variant is noted second. Thus we have: for some member $\mathbf{d}_j$ of the domain, $\mathrm{v_I}[\mathbf{d}_i/u, \mathbf{d}_j/v](G^2uv) = \mathbf{T}$. Now for each $\mathbf{d}_i$ there is a $\mathbf{d}_j$, and so the sentence is in fact true on I.

We are now in a position to state formally the semantical rules for *PI*.

Semantical Rules for *PI*

SR-TVA If α is a sentence-letter, then either $\mathbf{v_I}(\alpha)$=$\mathbf{T}$ or $\mathbf{v_I}(\alpha)$=$\mathbf{F}$; it is not the case that $\mathbf{v_I}(\alpha)$=$\mathbf{T}$ and $\mathbf{v_I}(\alpha)$=$\mathbf{F}$.

SR-$\bot$ For all I,$\mathbf{v_I}(\bot)$=$\mathbf{F}$ and $\mathbf{v_I}(\bot)\neq\mathbf{T}$.

[8] The same symbol, '=,' is used as both an object-language expression and as a meta-logical symbol for identity. Context will determine which usage is intended.

SR-Pred $\mathbf{v_I}(\mathbf{P}^n\mathbf{t}_1\ldots\mathbf{t}_n)=\mathbf{T}$ if and only if $\langle\mathbf{v_I}(\mathbf{t}_1),\ldots,\mathbf{v_I}\,(\mathbf{t}_n)\rangle \in \mathbf{v_I}(\mathbf{P}^n)$; $\mathbf{v_I}(\mathbf{P}^n\mathbf{t}_1\ldots\mathbf{t}_n)=\mathbf{F}$ if and only if $\neg(\langle\mathbf{v_I}(\mathbf{t}_1),\ldots,\mathbf{v_I}(\mathrm{t}_n)\rangle \in \mathbf{v_I}(\mathbf{P}^n))$

SR-∼ $\mathbf{v_I}(\sim\alpha)=\mathbf{T}$ if and only if $\mathbf{v_I}(\alpha)=\mathbf{F}$; $\mathbf{v_I}(\sim\alpha)=\mathbf{F}$ if and only if $\mathbf{v_I}(\alpha)=\mathbf{T}$.

SR-∧ $\mathbf{v_I}(\alpha\wedge\beta)=\mathbf{T}$ if and only if $\mathbf{v_I}(\alpha)=\mathbf{T}$ and $\mathbf{v_I}(\beta)=\mathbf{T}$; $\mathbf{v_I}(\alpha\wedge\beta)=\mathbf{F}$ if and only if $_\mathbf{I}(\alpha)=\mathbf{F}$ or $\mathbf{v_I}(\beta)=\mathbf{F}$.

SR-∨ $\mathbf{v_I}(\alpha\vee\beta)=\mathbf{T}$ if and only if $\mathbf{v_I}(\alpha)=\mathbf{T}$, or $\mathbf{v_I}(\beta)=\mathbf{T}$; $\mathbf{v_I}(\alpha\vee\beta)=\mathbf{F}$ if and only if $\mathbf{v_I}(\alpha)=\mathbf{F}$ and $\mathbf{v_I}(\beta)=\mathbf{F}$.

SR-⊃ $\mathbf{v_I}(\alpha\supset\beta)=\mathbf{T}$ if and only if $\mathbf{v_I}(\alpha)=\mathbf{F}$ or $\mathbf{v_I}(\beta)=\mathbf{T}$; $\mathbf{v_I}(\alpha\supset\beta)=\mathbf{F}$ if and only if $\mathbf{v_I}(\alpha)=\mathbf{T}$ and $\mathbf{v_I}(\beta)=\mathbf{F}$.

SR-≡ $\mathbf{v_I}(\alpha\equiv\beta)=\mathbf{T}$ if and only if either $\mathbf{v_I}(\alpha)=\mathbf{T}$ and $\mathbf{v_I}(\beta)=\mathbf{T}$, or $\mathbf{v_I}(\alpha)=\mathbf{F}$ and $\mathbf{v_I}(\beta)=\mathbf{F}$; $\mathbf{v_I}(\alpha\equiv\beta)=\mathbf{F}$ if and only if either $\mathbf{v_I}(\alpha)=\mathbf{T}$ and $\mathbf{v_I}(\beta)=\mathbf{F}$, or $\mathbf{v_I}(\alpha)=\mathbf{F}$ and $\mathbf{v_I}(\beta)=\mathbf{T}$.

SR-∀ $\mathbf{v_I}((\forall\mathbf{x})(\alpha(\mathbf{x}/\mathbf{u}))=\mathbf{T}$ (where $\mathbf{u}$ is free for $\mathbf{x}$ in $\alpha(\mathbf{x})$) if and only if for all $\mathbf{d}\in\mathbf{D}$, $\mathbf{v}_\mathrm{I}[\mathbf{d}/\mathbf{u}](\alpha(\mathbf{u}))=\mathbf{T}$; $\mathbf{v_I}((\forall\mathbf{x})(\alpha(\mathbf{x}/\mathbf{u}))=\mathbf{F}$ if and only if for some $\mathbf{d}\in\mathbf{D}$, $\mathbf{v_I}[\mathbf{d}/\mathbf{u}](\alpha(\mathbf{u}))=\mathbf{F}$.

SR-∃ $\mathbf{v_I}((\exists\mathbf{x})(\alpha(\mathbf{x}/\mathbf{u}))=\mathbf{T}$ (where $\mathbf{u}$ is free for $\mathbf{x}$ in $\alpha(\mathbf{x})$) if and only if for some $\mathbf{d}\in\mathbf{D}$, $\mathbf{v}_\mathrm{I}[\mathbf{d}/\mathbf{u}](\alpha(\mathbf{u}))=\mathbf{T}$; $\mathbf{v_I}((\exists\mathbf{x})(\alpha(\mathbf{x}/\mathbf{u}))=\mathbf{F}$ if and only if for all $\mathbf{d}\in\mathbf{D}$, $\mathbf{v}_\mathrm{I}[\mathbf{d}/\mathbf{u}](\alpha(\mathbf{u}))=\mathbf{F}$.

$\mathbf{v_I}(\mathbf{t}_i=\mathbf{t}_j)=\mathbf{T}$ if and only if $\mathbf{v_I}(\mathbf{t}_i)=\mathbf{v_I}(\mathbf{t}_j)$;
$\mathbf{v_I}(\mathbf{t}_i=\mathbf{t}_j)=\mathbf{F}$ if and only if $\mathbf{v_I}(\mathbf{t}_i)\neq\mathbf{v_I}(\mathbf{t}_j)$;

We will assert here without proof that the properties of *Quantificational Bivalence* and *Quantificational Truth-Functionality* hold in the semantical system *PI*. The semantical rules for the quantifiers are set up in such a way that all possible interpretations assign at least one truth-value, and no more than one, to each quantified sentence. The key move in the proofs would be that by the Inductive Hypothesis, the sentences containing parameters which form the basis for quantified sentences have one and only one truth-value. The notions of *Semantical Entailment in PI*, *Semantical Equivalence in PI*, *Validity in PI*, and *Semantical Consistency in PI* are carried over directly from the semantical system *SI*.

An example of semantical entailment in *PI* is this: $\{(\forall x)F^1x\}\vDash_{PI}F^1u$, a semantical form of universal instantiation. Suppose that for an arbitrary interpretation $\mathbf{I}$, $\mathbf{v_I}(\forall x)F^1x)=\mathrm{T}$. Then for every $\mathbf{d}$ in the domain of $\mathbf{I}$, $\mathbf{v}[\mathbf{d}/u]_\mathbf{I}(F^1u)=\mathbf{T}$. In that case, $\mathbf{v_I}(F^1u)=\mathbf{T}$, since $\mathbf{v_I}$ assigns some member $\mathbf{d}$ of the domain to 'u.'

An example of semantical consistency in *PI* is that the set '$\{(\exists x)F^1, \sim F^1a\}$ is semantically consistent. Consider an interpretation I with a domain consisting of the numbers 1 and 2, and let $\mathrm{v_I}$ assign the number 2 to 'a.' Finally, let 'F^1 be interpreted as the property of being odd. Then '$(\exists x)F^1x$' is true on I because there is a member of the domain that is odd. But 'F^1a is false in I, because the number assigned by v to 'a' is not odd. Hence '$\sim F^1a$' is true on I.

3 The Derivational System *PD*

The derivational system *PD* for Predicate Logic consists of four primitive rules for the quantifier, one for introducing and one for eliminating each of the two quantifiers. There are rules introducing and eliminating the identity symbol. Other rules function as derived rules in the system.

Before the rules can be stated, we need to extend our meta-logical terminology a bit. Thus far, we have used the notation $\alpha(\mathbf{x}/\mathbf{u})$ to indicate the result of substituting $\mathbf{u}$ for all occurrences of x in α. For the purposes of derivations, we will need to go in the opposite direction and be more general, beginning with a quasi-sentence '$\alpha(\mathbf{x})$' obtained from removing a quantifier and substituting a term '$\mathbf{t}$' (i.e., a parameter or a constant) for '$\mathbf{x}$.' Thus we will write '$\alpha(\mathbf{t}/\mathbf{x})$,' for some term '$\mathbf{t}$.' The result will be called a *substitution instance* of the quantified sentence. For example, 'F^1u' and 'F^1a' are a substitution instances of '$(\forall x)F^1x$.' The manipulation of substitution instances is central to the derivational rules for predicate logic.

3.1 Universal Quantifier Rules

3.1.1 Universal Elimination

The rule of Universal Elimination (∀E, also known as "Universal Instantiation") allows one to remove a universal quantifier and write down a substitution instance of the sentence it governs.

Universal Elimination

$$\begin{array}{|ll} (\forall \mathbf{x})\alpha(\mathbf{x}/\mathbf{u}) & \text{Already Derived} \\ \vdots & \\ \alpha(\mathbf{t}/\mathbf{u}) & \forall\ \mathrm{I} \end{array}$$

Instantiation may be made to a constant or parameter only, and not to a variable. It must also be uniform, in that there is only one instantiating term. All occurrences of the variable must be replaced with a term. (It is often essential to instantiate to the right term, in which case it is best to wait to see what term is required before instantiating.) We will give an example of the use of ∀ Elimination after stating the rule for ∀ Introduction.

Use of the rule of Universal Introduction will not lead us from truth to falsehood. If a universally quantified sentence $(\forall \mathbf{x})\alpha(\mathbf{x}/\mathbf{u})$ is true on an interpretation, then for all members $\mathbf{d}$ of its domain, $\alpha(\mathbf{u})$ will be true if $\mathbf{d}$ is assigned to $\mathbf{u}$. (We have already seen an instance of this relation: $\{(\forall x)F^1x\} \vDash_{PI} F^1u$.) Moreover, any sentence which results from the substitution of a constant will also be true, since each constant is assigned to at least one member of the domain. As will be seen in later modules, this latter condition does not obtain in some versions of Free Predicate Logic and Modal Predicate Logic based on Free Predicate Logic.

3.1.2 Universal Introduction

The rule of Universal Introduction (∀ I, also known as "Universal Generalization") allows one to replace all occurrences of a parameter (not a constant) with a variable and prefix a universal quantifier to the beginning of the resulting sentence. This is the first time that the role of a parameter differs from that of a constant. The rationale for using parameters is that we would like to have a way of indicating an *arbitrary* member of the domain. Universal Generalization is sound when something is shown to hold for a member of the domain without taking into account anything else about it. In that case, a given interpretation might assign to $\mathbf{u}$ any member of the domain at all, and so what is concluded about such an arbitrary member of the domain applies to all members of the domain.

To guarantee arbitrariness, we introduce a (non-modal) version of a restricted scope line. The line will be flagged by a parameter and will be treated as a "barrier" in the sense that no sentence containing that parameter may be reiterated across it. Because the parameter is "trapped," as it were, behind the barrier, no sentences containing it play any role in the results for it that are obtained within the confines of the barrier. (Keep in mind that no inference rules may be applied to sentences outside a restricted scope line: any previous sentence that might be involved in the use of a rule of inference must be first reiterated.)

Universal Introduction

$$
\begin{array}{|l}
\vdots \\
\begin{array}{r|l}
\mathbf{u} & \\
 & \vdots \\
 & \alpha(\mathbf{u})
\end{array} \\
(\forall \mathbf{x})\alpha(\mathbf{x}/\mathbf{u}) \qquad \forall\ \mathrm{I}
\end{array}
$$

Provided that: No sentence containing **u** is reiterated across the restricted scope line.

The following is an example of the use of the rule of Universal Generalization.

Proof that: $\{(\forall x)(Fx \wedge Gx)\} \vdash_{PD} (\forall y)Fy$

$$
\begin{array}{rll}
1 & \mid (\forall x)(Fx \wedge Gx) & \text{Assumption} \\
2 & \mid \; u \mid (\forall x)(Fx \wedge Gx) & \text{1 Reiteration} \\
3 & \mid \;\;\;\; \mid Fu \wedge Gu & 2\ \forall\ \mathrm{E} \\
4 & \mid \;\;\;\; \mid Fu & 3\ \&\ \mathrm{E} \\
5 & \mid (\forall y)Fy & 4\ \forall\ \mathrm{I}
\end{array}
$$

3.2 Existential Quantifier Rules

3.2.1 Existential Introduction

The rule of Existential Introduction (∃ I, also known as "Existential Generalization") allows one to replace any number of occurrences of a term with a free variable and prefix an existential quantifier to the beginning of the resulting sentence.

Existential Introduction

$$
\begin{array}{|ll}
\alpha(\mathbf{t}/\mathbf{u}) & \text{Already Derived} \\
\vdots & \\
(\exists \mathbf{x})\alpha(\mathbf{x}/\mathbf{u}) & \exists\ \mathrm{I}
\end{array}
$$

There are no restrictions on the generalization. One may generalize on one of the terms, more than one, or all of them.

The soundness of this rule depends on the fact that each term **t** is assigned by a given interpretation to a member **d** of the domain. If a sentence containing that term is true, then by the semantical rule for the '∃' operator, its existential quantification is true as well.

Here is an example of the use of the rule of Existential Generalization. Note that a parameter could have been used instead of a constant.

Proof that: $\{(\forall x)Fx\} \vdash_{PD} (\exists x)Fx$

1	$(\forall x)Fx$	Assumption
2	Fu	1 ∀ E
3	$(\exists y)Fy$	2 ∃ I

3.2.2 Existential Elimination

The rule of Existential Elimination (∃ E, also known as "Existential Instantiation") allows one to remove an existential quantifier from $(\exists \mathbf{x})\alpha(\mathbf{x})$, replacing it with a substitution instance $\alpha(\mathbf{a}/\mathbf{x})$, made with a constant **a** not currently used, within a new assumption. A sentence β not containing the constant **a** is derived from that assumption, and the assumption is discharged, with the sentence β brought out.

Existential Elimination

	$(\exists \mathbf{x})\alpha(\mathbf{x}/\mathbf{u})$	Already derived
$\mathbf{u}$	$\alpha(\mathbf{u})$	Assumption
	$\vdots$	
	β	
	β	∃ E

Provided that: no sentence containing **u** is reiterated across the restricted scope line,
u does not occur in β.

The soundness of this rule is more complicated to explain. We suppose that some existentially quantified sentence is true on an interpretation. Then we assume that it is true for one of its substitution instances, where a parameter is substituted for the variable and the quantifier removed. That the assignment to this parameter is arbitrary is guaranteed by the fact that no sentence containing it can be reiterated. If a sentence β not containing the parameter is derived, the role of the parameter has been exhausted, so to speak, and the sentence is true.

The following are two examples of the use of Existential Elimination.

To prove: $\{(\exists x)Fx\} \vdash_{PD} (\exists y)Fy$

1		$(\exists x)Fx$	Assumption
2	u	Fu	Assumption
3		$(\exists y)Fy$	2 ∃ I
4		$(\exists y)Fy$	1 2-3 ∃ E

To prove: $\{(\exists x)(Fx \wedge Gx)\} \vdash_{PD} (\exists y)Fy \wedge (\exists z)Gz$

1	$(\exists x)(Fx \wedge Gx)$	Assumption
2	$u \mid Fu \wedge Gu$	Assumption
3	$\mid Fu$	2 ∧ E
4	$\mid Gu$	2 ∧ E
5	$\mid (\exists y)Fy$	3 ∃ I
6	$\mid (\exists z)Gz$	4 ∃ I
7	$\mid (\exists y)Fy \wedge (\exists z)Gz$	5 6 ∧ I
8	$(\exists y)Fy \wedge (\exists z)Gz$	1 2-7 ∃ E

3.3 Identity Rules

Here we give rules for introducing and eliminating sentences containing the identity sign.

3.3.1 Identity Introduction

The rule for introducing an identity-sentence is unique, in that it allows us to write down a sentence without any regard from what has come before it. Thus, it functions in the manner of an axiom.

Identity Introduction

$\vdots$	
$\mathbf{t} = \mathbf{t}$	= I

In the semantics, an interpretation always assigns a term **t** to a member of the domain **t**. The ordered pair $\langle \mathbf{v_I}(\mathbf{t}), \mathbf{v_I}(\mathbf{t}) \rangle$ is guaranteed to be in the extension of '=.'

3.3.2 Identity Elimination

The rule for eliminating an identity sentence allows for the substitution, in another sentence, of a term $\mathbf{t}_i$ that occurs on one side of it for the term $\mathbf{t}_j$ that occurs on the other side. The order of occurrence of the identity sentence and the sentence in which the substitution is made does not affect the use of the rule. Also, the identity-sentence and the substitution-sentence may be the same.

Identity Elimination

$\alpha(\mathbf{t}_i)$	Already Derived
$\vdots$	
$\mathbf{t}_i = \mathbf{t}_j$	Already Derived
$\vdots$	
$\alpha(\mathbf{t}_j/\mathbf{t}_i)$	= E

This rule is sound because if both $\mathbf{t}_i$ and $\mathbf{t}_j$ are assigned the same member **d** of the domain, as would be the case if the identity is true on an interpretation, then the way in which $\alpha(\mathbf{t}_i)$ is interpreted will be exactly the way $\alpha(\mathbf{t}_j)$ is interpreted, since $\mathbf{v_I}(\mathbf{t}_i) = \mathbf{v_I}(\mathbf{t}_j)$.

Here is an example of both rules at work.

To prove: $\vdash_{PD} \{(\forall x)(\forall y)(x = y \supset y = x)$

1	u	v	$u = v$	Assumption
2			$u = u$	= I
3			$v = u$	1 2 = E
4			$u = v \supset v = u$	1-3 ⊃ I
5			$(\forall y)(u = y \supset y = u)$	4 ∀ I
6			$(\forall x)(\forall y)(x = y \supset y = x)$	5 ∀ I

3.4 Derived Rules

Rules of *Quantifier Exchange* are easily derived within *PD*. The rules will first be stated, and then the derivation of the first rule will be given. Derivation of the other rules will be left as an exercise. Note that in sentences of the form $\alpha(\mathbf{x}/\mathbf{u})$, the parameter **u** does not occur, as it has been replaced by **x**. Hence, there is no problem with reiterating such sentences (or sentences of which they are components) across restricted scope lines.

Quantifier Exchange

$\sim(\forall \mathbf{x})\alpha(\mathbf{x}/\mathbf{u})$	Already Derived
$(\exists \mathbf{x})\sim\alpha(\mathbf{x}/\mathbf{u})$	~∀

$\sim(\exists \mathbf{x})\alpha(\mathbf{x}/\mathbf{u})$	Already Derived
$(\forall \mathbf{x})\sim\alpha(\mathbf{x}/\mathbf{u})$	~∃

$(\exists \mathbf{x})\sim\alpha(\mathbf{x}/\mathbf{u})$	Already Derived
$\sim(\forall \mathbf{x})\alpha(\mathbf{x}/\mathbf{u})$	∃~

$(\forall \mathbf{x})\sim\alpha(\mathbf{x}/\mathbf{u})$	Already Derived
$\sim(\exists \mathbf{x})\alpha(\mathbf{x}/\mathbf{u})$	∀~

To prove: $\{\sim(\forall \mathbf{x})\alpha(\mathbf{x}/\mathbf{u})\} \vdash_{PD} (\exists \mathbf{x})\sim\alpha(\mathbf{x}/\mathbf{u})$

```
│ ~(∀x)α(x/u)                  Already Derived
│ │ ~(∃x)~α(x/u)               Assumption
│ ├──
│ │u│ │ ~α(u)                  Assumption
│ │ │ ├──
│ │ │ │ (∃x)~α(x/u)            ∃ I
│ │ │ │ ~(∃x)~α(x/u)           Reiteration
│ │ │ ~α(x/u)                  ~ E
│ │ (∀x)α(x/u)                 ∀ I
│ │ ~(∀x)α(x/u)                Reiteration
│ (∃x)~α(x/u)                  ~ E
```

The reader might note the structuraly similarity between the Quantifier Exchange rules and the Duality rules for derivations in Modal Sentential Logic. This similarity has a semantical underpinning. In the meta-logic we quantify over possible worlds, and this quantification is reflected in the semantical behavior of the modal operators. The most straightforward case is with the semantical system *S5I* without the accessibility relation. The claim in the meta-language that at all worlds, it is not the case that α is true (the condition for the truth of '$\Box\sim\alpha$,' is equivalent to the claim that there is no world at α is true, which is the condition for the truth of '$\sim\Diamond\alpha$.'