HANDOUT #6 – PREDICATE LOGIC – SYMBOLS, SYNTAX, SEMANTICS, TRANSLATION

Predicate Logic: Introduction

The language of PL has three principal strengths:

> (S1) for any argument that is valid in PL, there is a corresponding valid argument in English. This means that some of the arguments that we wish to represent and the reasoning we do in English can be represented in the more precise language of PL.
> (S2) There are decision procedures for PL! We can use *tables* and *trees* to mechanically check whether an argument is valid or invalid in PL, whether a set of propositions are consistent or inconsistent, etc.
> (S3) There is a proof system (PD) for PL. That is, we have a codified set of rules that justify various derivations or moves forward in arguments.

The principal weakness of PL is the following:

> (W1) It is *not expressive* enough. That is, there are some valid arguments in English that cannot be represented in PL. For example,
>
> > All men are mortal.
> > Socrates is a man.
> > Socrates is mortal.

The reason that the above argument cannot be represented in PL is that PL treats sentences as *wholes*. The validity of the above argument, however, depends upon how *parts* of the sentences relate to each other, namely how Socrates belongs to a class of things (i.e. men) and how the class of men form a part of a more expansive class, namely mortals. In order to capture how parts of sentences relate to each other, we can devise a more expressive logical language, one that analyzes sentences at the *sub-sentential* level. This is the language of predicate logic (or the logic of relations): RL.

	Description of the Symbols of the RL	**Symbols of RL**
1	individual constants (names)	Lower case letters, *a* through *v* with or without numerical subscripts.
2	n-place predicates	Upper case letters, A through Z with or without numerical subscripts
3	individual variables	Lower case letters, *w* through *z* with or without numerical subscripts
4	truth-functional operators and scope indicators from PL	$\neg, \wedge, \vee, \rightarrow, \leftrightarrow$, (,), [,], {, }
5	Quantifiers	$\forall, \exists$

While we will talk more about what each of these items means (semantics), the ways in which they can be put together to form wffs (syntax), and how to translate from predicate logic into English, here is a very simple way to think about each:

> **individual constants** (a, b, c, d, e) are like *proper names*, they refer to specific objects. For example, *George Washington, Hurricane Sandy, Gandhi.*
> **n-place predicates** (A, B, C, D, E) are like *predicate terms* that refer to properties that belong to objects. For example, *is blue, is happy, is taller than, etc.*
> **individual variables** are like placeholders for objects, e.g. in "*x* is happy" x is a placeholder for some name we could insert to make that statement true or false.
> **quantifiers** are just terms that specifying the quantity of the objects that have some property. For example, *Some, All, Many, At least one x is green*.

Predicate Logic: Syntax (Formation Rules)

> (i) An *n*-place predicate **P** followed by *n* terms (names or variables) is a wff in RL.
> (ii) If **P** is a wff in $\mathbb{RL}$, then **¬P** is a wff in RL.
> (iii) If **P** and **Q** are wffs in $\mathbb{RL}$, then **P∧Q, P∨Q, P→Q,** and **P↔Q** are wffs in RL.
> (iv) If **P** is a wff in $\mathbb{RL}$ containing a name **a** and if **P(x/a)** is what results from substituting the variable **x** for every occurrence of **a** in **P**, then (∀x)**P(x/a)** and (∃x)**P(x/a)** are wffs in RL.
> (v) Nothing else is a wff in $\mathbb{RL}$ except that which can be formed by repeated applications of (i)–(iv).

The above rules can be used to determine whether a formula is a wff. Here is an example:

Example: show that (∀z)¬Pz∧¬Ra is a wff in RL

1	Ra is a wff	Rule (i)
2	Pa is a wff	Rule (i)
3	If Ra is a wff, then ¬Ra is a wff.	Line 1, rule (ii)
4	If Pa is a wff, then ¬Pa is a wff.	Line 2, rule (ii)
5	If ¬Pa is a wff in RL and ¬Pz is what results from replacing *a's* with *z's*, then (∀z)¬Pz is a wff.	Line 4, Rule (iv)
6	If ¬Ra is a wff and ¬(∀z)Pz is a wff, then (∀z)¬Pz∧¬Ra is a wff in RL.	Line 3, 5, rule (iii)

Classroom Exercises*:* Using the formation rules, show that the following propositions are wffs in $\mathbb{RL}$, where 'Pxy' is a two-place predicate while 'Rx' and 'Zx' are one-place predicates:
1. Ra∧Paa
2. Raa→Pa
3. (∀x)Pxx
4. (∃x)Px
5. ¬(∃y)Pyy
6. ¬(∀x)Pxx∧(∃x)Zx

More Syntax: Quantifier Scope and Further Terminology

The scope of the quantifiers is similar to the scope of the truth-functional operator for negation. That is, it applies to the wff (open or closed) to its immediate right.

$(\forall x)Px$

However, there is some ambiguity when the formula upon which quantifiers operate are complex. Consider the following:

$(\forall x)Px \rightarrow Rx$

Does the quantifier apply to 'Px' alone or does it apply to the complex '$Px \rightarrow Rx$'? In order to resolve this ambiguity, scope indicators (open and closed parentheses, braces, and brackets) are used to indicate the scope of the quantifier. Thus, in the above example, the universal quantifier applies only to 'Px' while in the following formula, it operates upon '$Px \rightarrow Rx$':

$(\forall x)(Px \rightarrow Rx)$

Quantifiers can have other quantified formula in their scope. For example, in the following formula, the existential quantifier has the formula to its right in its scope, i.e. '$(\forall y)(Px \rightarrow Ry)$'

$(\exists x)(\forall y)(Px \rightarrow Ry)$

Quantifiers specify the quantity of specific variables. For example, in the above formula, '$(\exists x)$' quantifies '*xs*' while '$(\forall y)$' quantifies '*ys*'. We call a variable that is quantified by a quantifier that quantifies for that specific variable, a **bound variable**. A variable that is not bound is a **free variable**. We call a variable is in the scope of a quantifier, irrespective of whether the quantifier quantifies for it, a **scoped variable**.

Examples:
$(\forall x)Fx$, x is bound and scoped
$(\forall x)Rxy$, x is bound and scoped but y is free and scoped
$(\exists x)(Rx \wedge Lx)$, both xs are bound and scoped

Open formula	An *open formula* is a wff consisting of an n-place predicate **P** followed by *n* terms where one of those terms is a free variable
Closed formula	A *closed formula* is a wff consisting of an n-place predicate **P** followed by *n* terms where every term is a name or a bound variable.

Classroom Exercises: *Determine whether the formula is open or closed and whether variables are bound, scoped, and/or free.*
1. $(\forall x)Pxy$
2. $(\exists x)(\forall y)(\forall z)(Pxy \rightarrow Lz)$
3. $(\exists x)(\forall y)(\forall z)Pxy \rightarrow Lz$

Semantics: The Absolute Basics

There is a long discussion of the semantics of RL in your text, but here are the absolute basics.

> In the semantics of PL, single propositional letters are assigned truth values (T or F). This is not possible in RL since individual constants (names), predicate terms, and quantifiers are not true or false.

The first key semantic concept is the **domain of discourse** (abbreviated as '$\mathfrak{D}$') which are all of the things we want to talk about. For example, numbers, physical objects, favorite movies, etc. We indicate what objects are in the domain by either listing them like as follows:

$\mathfrak{D}$ = {Jon, Vic, Liz, Tek}

Or, indicating some property they all have in common:

$\mathfrak{D}$ = {x | x is a living human being}.

The above is read as "the domain consists of all beings that have the property of being a living human being."

Next, there is what is called an **interpretation** (abbreviated as '$\mathscr{I}$'). An interpretation does three things:

(i) it assigns objects in the domain to names in RL:

Example of (i): $\mathscr{I}$(a): Liz; $\mathscr{I}$(b): Vic; $\mathscr{I}$(c): Jon; $\mathscr{I}$(c): Tek

(ii) it assigns sets of objects or sets of ordered n-tuples in the domain to predicates in RL:

Example of (ii): $\mathscr{I}$(Hx): {Jon, Liz, Vic}
You can think of 'Hx' as 'x is happy', an interpretation of 'x is happy' picks out all of the happy objects.

Example of (ii): $\mathscr{I}$(Lxy): {⟨Jon, Liz⟩, ⟨Liz, Vic⟩}
You can think of 'Lxy' as 'x loves y', an interpretation of 'Lxy' picks out pairs of objects that are ordered in a certain way. Since *Jon loves Liz* is different than *Liz loves Jon*, we need a way of representing the former that cannot be confused with the latter.

Jon loves Liz: ⟨Jon, Liz⟩
Liz loves Jon: ⟨Liz, Jon⟩

(iii) it assigns truth values to closed formula in RL:

Example of (iii): v(Lca)=T; *v*(Lac)=T; *v*((∀x)Lxa)=F

Truth values are assigned to *non-quantified formula* as follows: a closed formula consisting of a predicate **P** followed by the appropriate number of names $\mathbf{a_i}$ is true if and only if the interpretation of $\mathbf{a_i}$ is in **P**.

What does this mean?
Example #1: 'Hx' picks out a set of things. The interpretation of 'Hx' selects Jon, Liz, and Vic. The interpretation of 'a' selects *Liz*. Since the interpretation of 'a' is in the interpretation of 'Hx', then 'Ha' is true.
Example #2: 'Lab' is true if and only if the interpretation of ⟨a, b⟩ is in the interpretation of L. 'Lxy' picks out a set of pairs of things: {⟨Jon, Liz⟩, ⟨Liz, Vic⟩}; since the interpretation of '⟨a, b⟩' refers to the pair ⟨Liz, Vic⟩, and this is in the set of things picked out by 'Lxy' then 'Lab' is true.

Truth values are assigned to *quantified formula* (very roughly, see text) as follows:
Example #1: '(∀x)Px' is true if every object in the domain is P.
Example #2: '(∃x)Px' is true if and only if at least one object in the domain is P.

Classroom Exercises: RL Semantics

Practice
Let D = {a, b, c}, $\mathscr{I}$(a)=a, $\mathscr{I}$(b)=b, $\mathscr{I}$(c)=c, $\mathscr{I}$(Hx)= {a, b, c}, $\mathscr{I}$(Lxy)= {⟨a, a⟩, ⟨b, c⟩}
1. Ha
2. (∃x)Hx
3. (∀x)Hx
4. ¬(∀x)Hx
5. (Ha∧Hb)∧Hc

1. Lab
2. ¬Laa
3. Lba
4. (∃x)Lxx
5. (∀x)Lxx
6. (∃x)Lxx∨(∃x)(∃y)Lxy

Predicate Logic: Translation

The formal language of RL can be used to express a fragment of the English language. The first step to any translation is to construct a translation key. A **translation key** does three things: (1) it stipulates the domain of discourse, (2) it interprets all names and (3) interprets all n-place predicates. Consider the following key.

> $\mathfrak{D}$: Living human beings
> j: Jon
> f: Frank
> Txy: x is taller than y

Using the above translation key we can translate various English sentences into RL and closed formula in RL into English sentences.
Here is a quick guide:

Predicate logic formula	*Read in pseudo-English as…*
Ha	a is H.
Tab	a is taller than b
(∃x)Hx	Some x is H. At least one x is H.
(∀x)Hx	Every x is H.
(∀x)(∃y)Txy	Every x is taller than y.

Note that '(∀x)' can be used to translate pseudo-English expressions like 'for every *x*', 'for all *x*s', 'for each *x*'. Likewise '(∃x)' can be used to translate pseudo-English expressions like 'for some *x*', 'some *xs*', or 'there exists an *x*'.

Using the above translation key we can translate the following English sentences:

> (1) Jon is taller than Frank.
> (2) Frank is taller than Jon.
> (3) It is not both the case that Jon is taller than Frank and Frank is taller than Jon.

into RL as follows:

> (1*) Tjf
> (2*) Tfj
> (3*) ¬(Tjf ∧ Tfj)

The method for translating English expressions into quantified formula in RL is an *art* and so it can help to take a step-by-step method. One way to do this is to create a *bridge translation* between wffs in RL and sentences in English, and then use this bridge translation to translate into more colloquial English. A bridge translation is a *half-way point* between English and RL; it's not quite English and not quite RL: it's *pseudo-English*.

Consider the following translation key

$\mathfrak{D}$: human beings (living or dead)
Hx: x is happy
Zx: x is a zombie
Mx: x is mortal
Rx: x is murderer
Wx: x is wrong

Now consider the following predicate wffs:

(1) $(\forall x)Hx$
(2) $(\forall x)\neg Zx$
(3) $(\forall x)(Zx\rightarrow Hx)$
(4) $(\forall x)(Zx\rightarrow\neg Hx)$
(5) $\neg(\forall x)(Zx\rightarrow Hx)$

Let's consider a translation of (1) by taking one part of the formula at a time. '$(\forall x)$' is translated as 'For every *x*', 'for all *x*s', 'for each *x*'. The second part of (1) says '*x* is H' or '*x* is happy'. Putting these two parts together we get a *bridge translation*. Here is a bridge translation of (1):

(1_B) For every *x*, *x* is happy.

Using this bridge translation, we can more easily translate (1) into colloquial English:

(1_E) Everyone is happy.

Consider a bridge translation of (2):

(2_B) For every *x*, *x* is not a zombie.

Using (2_B) we can render (2) into something more natural:

(2_E) Everyone is a not a zombie.
(2_{E*}) No one is a zombie.

Consider a bridge translation of (3):

(3_B) For every *x*, if *x* is a zombie, then *x* is happy.

In the case that (3_B) does not make it obvious how to render it into English, then you can try to make (3_B) more concrete by expanding the bridge translation as follows:

(3_{B+}) Choose any object you please in the domain of discourse, if that object is a zombie, then it will be also be happy.

Rendered into standard English, (3_B) and (3_{B*}) reads:

(3_E) Every zombie is happy.

Consider a bridge translation of (4):

(4_B) For every x, if x is a zombie, then x is not happy.

An additional bridge translation is the following:

(4_{B*}) Choose any object you please in the domain of discourse consisting of human beings (living or dead), if that object is a zombie, then it will not be happy.

In colloquial English this is the following:

(4_E) No zombies are happy.

Notice that in the case of (5), which is (5) $\neg(\forall x)(Zx\rightarrow Hx)$, the main operator is *negation*. One way to translate this by first translating '$(\forall x)(Zx\rightarrow Hx)$':

Every zombie is happy.

Next, translate the negation into English by putting 'Not' in front of this expression. That is, (5) reads:

(5_E) Not every zombie is happy.

Finally, consider universally quantified expressions not involving $\rightarrow$ as the main operator

(6) $(\forall x)(Zx\wedge Hx)$
(7) $(\forall x)(Zx\vee Hx)$
(8) $(\forall x)(Zx\leftrightarrow Hx)$

These are

(6_E) Everyone is a happy zombie.
(7_E) Everyone is either a zombie or happy.
(8_E) Everyone is a zombie if and only if they are happy.

Our focus thus far has been on the use of the universal quantifier to translate RL formula into English. Next, we turn to RL formulas that involve the existential quantifier. Consider the following predicate wffs:

(1) $(\exists x)Hx$
(2) $(\exists x)\neg Zx$
(3) $\neg(\exists x)Zx$
(4) $(\exists x)(Zx \wedge Hx)$
(5) $(\exists x)Zx \wedge (\exists x)Hx$

Let's consider a translation of (1) by taking one part of the formula at a time. (∃x) is translated as 'For some *x*', 'there exists an *x*', 'there is at least one *x*'. The second part of (1) says '*x* is H' or '*x* is happy'. Putting thse two parts together we get a *bridge translation*. Again, a bridge translation is not quite English and not quite predicate logic. Here is a bridge translation of (1)

(1_B) For some *x*, *x* is happy.

(1) says that there is at least one in the object in the $\mathfrak{D}$ that has the property of being happy. Using the bridge translation (1_B), we can more easily translate (1) into colloquial English:

(1_E) Someone is happy.

Consider (2). Again, we can use a bridge translation,

(2_B) For some *x*, *x* is not a zombie.

(2_B) can be translated into colloquial English as follows:

(2_E) Someone is not a zombie.

In the case of (3), note that negation has wide scope. Thus, we can translate '(∃x)Zx' first, and then translate '¬(∃x)Zx'. That is, (∃x)Zx translates into 'Someone is a zombie', and ¬(∃x)Zx translates as:

(3_E) It is not the case that someone is a zombie.

Notice that (2) and (3) say something distinct. (2) says that something exists that is not a zombie, while all (3) says is that zombies do not exist. Let's consider (4) and (5) together. The bridge translations for (4) and (5) are as follows:

(4_B) For some x, x is a zombie *and* x is happy.
(5_B) For some x, x is a zombie, and for some x, x is happy.

Notice that these two propositions do not say the same thing. (4) asserts that there is something that is *both* a zombie and happy, while (5) asserts that there is a zombie *and* there is someone who is happy.

Finally, consider some propositions where ∧ is not the main operator.

(6) $(\exists x)(Zx \rightarrow Hx)$
(7) $(\exists x)(Zx \vee Hx)$
(8) $(\exists x)(Zx \leftrightarrow Hx)$

The bridge translations for these are

(6_B) For some x, if x is a zombie, then x is happy.

(7_B) For some x, x is a zombie or x is happy.
(8_B) For some x, x is a zombie if and only if x is happy.

and these can be translated into the following English expressions

(6_E) There exists something such that if it is a zombie, then it is happy.
(7_E) Someone is either a zombie or happy.
(8_E) Something is a zombie if and only if it is happy.

Notice that (6_E) is a bit strange. We might exploit the fact that $Zx \rightarrow Hx$ is equivalent to $\neg Zx \vee Hx$ and translate (6+) instead:

(6) $(\exists x)(\neg Zx \vee Hx)$
(6_{E+}) Something is either happy or not zombie.

Classroom Exercises: Basic Translation in RL

Key: $\mathcal{D}$: people
Px: x is poor
Lx: x is lazy
Rx: x is rich

Universal Quantifier
1. $(\forall x)Px$
2. $(\forall x)(Px \rightarrow Lx)$
3. $(\forall x)Px \wedge (\forall x)Lx$, what is the difference between #3 and #2?
4. $(\forall x)(Px \wedge Lx)$
5. $(\forall x)(Px \rightarrow \neg Lx)$
6. $(\forall x)(Px \vee Lx)$

Existential Quantifier
1. $(\exists x)Px$
2. $(\exists x)Px \wedge (\exists x)Rx$
3. $(\exists x)(Px \wedge Rx)$, what is the difference between #3 and #2?
4. $(\exists x)(Px \vee Rx)$
5. $\neg(\exists x)(Px \wedge Rx)$
6. $(\exists x)\neg(Px \wedge Rx)$

English to RL
1. All poor people are lazy.
2. All lazy people are poor.
3. Not all lazy people are poor.
4. Some lazy person is not poor.
5. Someone is lazy and someone is poor.
6.* If not all lazy people are poor, then not all poor people are lazy.

Semantics: A Longer Discussion (Optional)

In the RL, the basic unit of expression, the proposition, was capable of being true or false. This meant that the semantics of RL just involved assigning truth values to the well-formed formulas of this language. This cannot be the case for RL since the basic units of expressions do not expression propositions (something capable of being true or false) but are constituents (parts) of propositions. The semantics of RL has three key elements:

(1) A model, which consists of
(2) an interpretation, and
(3) a domain

Model	A *model* in $\mathbb{RL}$ is a structure consisting of a domain $\mathfrak{D}$ and an interpretation function $\mathscr{I}$.
Interpretation	An *interpretation* is an assignment of (i) objects in $\mathfrak{D}$ to names, (ii) a set of n-tuples in $\mathfrak{D}$ to n-place predicates, and (iii) truth values to sentences.
Domain	The *domain of discourse* (abbreviated as '$\mathfrak{D}$') consists of all of the things that a language can meaningfully refer to or talk about.

We give meaning to the names, n-place predicates, and wffs in RL by (i) assigning objects to names, (ii) assigning sets of objects (or sets of pairs, or sets of triples) to n-place predicates, and (iii) assigning truth values to closed formula.

What was somewhat glossed over (above) were interpretations of n-place predicates where n>1. Consider the following two-place predicate:

Lxy: x loves y

Here the interpretation does not simply tell us about a collection of single objects, i.e. about which objects which love or are loved. Instead, an interpretation of this two-place 'Lxy' tells us something about *pairs of objects*, i.e. about which objects love which objects. It tells us *who* loves *whom*. Thus, rather than saying an interpretation of an *n*-place predicate tells us something about a collection of objects, we say that it tells us something about a set (or collection) of *n-tuples*, where an **n-tuple** is a sequence of n-objects. In the above case, an interpretation of 'Lxy' is an assignment of a set of 2-tuples to 'Lxy.'

For convenience, we will write tuples by listing the elements (objects) within angle brackets '⟨ ⟩' and separate each element by commas. For example,

⟨Liz, Jon⟩

denotes a 2-tuple, while

⟨Liz, Vic, Jon⟩

denotes a 3-tuple.

Thus, suppose that in $\mathfrak{D}$, Jon loves Liz (and no one else), Liz loves Vic (and no one else), and Vic loves no one. The interpretation of 'Lxy' in $\mathfrak{D}$ would be represented by the following set of 2-tuples:

$\mathscr{I}$(Lxy): {⟨Jon, Liz⟩, ⟨Liz, Vic⟩}

This says that the interpretation of the predicate 'x loves y' relative to $\mathfrak{D}$ consists of a set with two 2-tuples, one 2-tuple is ⟨Bill, Corinne⟩ and the other is ⟨Corinne, Alfred⟩.

Third, an interpretation assigns truth values to closed formula.

1 $v(\mathbf{Ra_i})$=T if and only if (iff) the interpretation of $\mathbf{a_i}$ is in **R**, i.e. $\mathbf{a_i} \in \mathbf{R}$.
2 $v(\neg\mathbf{P})$=T iff $v(\mathbf{P})$=F
$v(\neg\mathbf{P})$=F iff $v(\mathbf{P})$=T
3 $v(\mathbf{P} \wedge \mathbf{Q})$=T iff $v(\mathbf{P})$=T, and $v(\mathbf{Q})$=T
$v(\mathbf{P}\wedge\mathbf{Q})$=F iff $v(\mathbf{P})$=F or $v(\mathbf{Q})$=F
4 $v(\mathbf{P} \vee \mathbf{Q})$=T iff either $v(\mathbf{P})$=T or $v(\mathbf{Q})$=T
$v(\mathbf{P}\vee\mathbf{Q})$=F iff $v(\mathbf{P})$=F and $v(\mathbf{Q})$=F
5 $v(\mathbf{P} \rightarrow \mathbf{Q})$=T iff either $v(\mathbf{P})$=F or $v(\mathbf{Q})$=T
$v(\mathbf{P}\rightarrow\mathbf{Q})$=F iff $v(\mathbf{P})$=T and $v(\mathbf{Q})$=F
6 $v(\mathbf{P}\leftrightarrow\mathbf{Q})$=T iff either $v(\mathbf{P})$=T and $v(\mathbf{Q})$=T, or $v(\mathbf{P})$=F and $v(\mathbf{Q})$=F
$v(\mathbf{P}\leftrightarrow\mathbf{Q})$=F iff either $v(\mathbf{P})$=T and $v(\mathbf{Q})$=F, or $v(\mathbf{P})$=F and $v(\mathbf{Q})$=T
7 $v(\forall x)\mathbf{P}$=T iff for *every* name **a** not in **P** and every **a-variant interpretation**, P(a/x)=T**.**
$v(\forall x)\mathbf{P}$=F iff for *at least one* **a** not in **P** and at least one **a-variant interpretation**, P(a/x)=F**.**
8 $v(\exists x)\mathbf{P}$=T iff for *at least one* name **a** not in **P** and at least one **a-variant interpretation**, P(a/x)=T**.**
$v(\exists x)\mathbf{P}$**=F** iff for *every* name **a** not in **P** and *every* **a-variant interpretation**, P(a/x)=F**.**

Much of the above is straightforward and similar to the semantics of PL. However, what is unique to RL concerns (1), (7), and (8).

The valuation of wffs with involving predicate terms and names is straightforward. A formula **Rai** is true if and only if **ai** is in the set of objects (or set of n-tuples) designated by **R**. For example, 'Jon is tall' is true if and only if Jon is in the set of tall objects. Likewise, 'Liz loves Jon' is true if and only if the 2-tuple ⟨Liz, Jon⟩ is in the interpretation of the 'Lxy'.

The valuations for (7) and (8) are done by defining truth values of quantified formulas by relying upon the truth values of simpler, non-quantified formula. This method requires a little care for, at least initially, we might say that '(∃x)Px' is true if and only if 'Px' is true given some replacement of 'x' with an object or a name (object constant) is true. Likewise, a wff like '(∀x)Px' is true if and only if 'Px' is true given that every replacement of 'x' with an object or name yield a true propositoin. However, this will not work without some further elaboration. On the one hand, we cannot replace variables with *objects from the domain* since variables are linguistic items and replacing a variable with an object won't be a wff. It won't even be a proposition. On the other hand, we cannot replace variables with *names from our logical vocabulary* because this falsely assumes that we have a name for every object in the domain. It might be the case that some objects in the domain are *unnamed*.

The solution to this problem is not simply expanding our logical vocabulary so that there is a name for every object but to consider the multitude of different ways in which a single *name* can be interpreted relative to the domain. The general idea is that '(∀x)**P**' is true if and only if

‘**P**(**a**/x)’ is true for every way of interpreting ‘**a**’ and ‘(∃x)**P**’ is true if and only if ‘**P**(**a**/x)’ is true for every way of interpreting ‘**a**’. To put this more precisely, consider that there are many different ways that an object in the domain can be assigned to a name. To see this more clearly, consider the following domain:

$\mathfrak{D}$: {Jon, Vic, Liz}

Now let’s consider the following interpretation $\mathscr{I}$ of ‘a’ relative to D:

$\mathscr{I}$(a): Jon

This is a perfectly legitimate interpretation, but we might think of a *variant interpretation* of $\mathscr{I}$, e.g.

$\mathscr{I}_1$(a): Vic

Further, we might even think of another *variant interpretation* of $\mathscr{I}$, e.g.

$\mathscr{I}_2$(a): Liz

Let’s say that for any name ‘**a**’, an interpretation $\mathscr{I}_{\mathbf{a}}$ is ‘**a-variant**’, ‘**a**-varies’, or is an ‘**a-variant interpretation**’ if and only if $\mathscr{I}_{\mathbf{a}}$ interprets ‘**a**’ (i.e. it assigns it an object in $\mathfrak{D}$) and it either does not differ from $\mathscr{I}$ or it differs only in the interpretation it assigns to ‘**a**’ (i.e. it doesn’t differ on the interpretation of any other feature of $\mathbb{RL}$). Thus, $\mathscr{I}$, $\mathscr{I}_1$, and $\mathscr{I}_2$ are all a-variant interpretations of $\mathscr{I}$ since they all assign ‘a’ to an object in the domain and they either do not differ from $\mathscr{I}$ or differ only in the interpretation they assign to ‘a’. We might then replace our intuitive definition of when ‘‘(∀x)**P**’ is true from

‘(∀x)**P**’ is true if and only if ‘**P**(**a**/x)’ is true for every way of interpreting ‘**a**’

to

‘(∀x)**P**’ is true iff for every a-variant interpretation, it is the case that the **P(a/x)** is true.

However, we need to make one further caveat. We said that for any name ‘**a**’, an interpretation $\mathscr{I}_{\mathbf{a}}$ is ‘**a-variant**’, ‘**a**-varies’, or is an ‘**a-variant interpretation**’ if and only if $\mathscr{I}_{\mathbf{a}}$ interprets ‘**a**’ (i.e. it assigns it an object in $\mathfrak{D}$) and it either does not differ from $\mathscr{I}$ or it differs only in the interpretation it assigns to ‘**a**’. Part of the idea here is that we want to *hold constant* our interpretation of all other formula (e.g. n-place predicate terms or other names) and consider the various ways of interpreting ‘**a**’ relative to the domain. For this to occur, the substituted name ‘**a**’ should not already occur in **P** since this name presumably already has an interpretation. In other words, determining the truth value of ‘(∀x)Pxa’ requires us to hold our interpretation of ‘P’ and ‘a’ fixed, and to cash out the truth value of quantified part of the formula by considering the various ways (the variant interpretations) a substituted name (other than ‘a’) could be interpreted in the domain.

Thus, we need to replace our revised definition of when ‘(∀x)P’ is true from

‘(∀x)**P**’ is true iff for every a-variant interpretation, it is the case that the **P(a/x)** is true.

to

v(∀x)**P**=T iff for *every* name **a** not in **P** and every **a-variant interpretation**, P(a/x)=T**.**

Using the notion of a variable interpretation, we have a solution to the problem of defining quantified formula by replacing variables with *names from our logical vocabulary* since our solution does not falsely assume that we have a name for every object in the domain. Instead, it assumes that there are many ways in which a name can be interpreted, and so while there may not always be a name for every object in the domain, there is always at least one variant interpretation that assigns a name to the previously unnamed object. Using this notion, a universally quantified proposition (∀x)**P** is true iff for every name **a** not in **P** and every a-variant interpretation, it is the case that the **P(a/x) is true**. In other words, it is true for every formula that is the result of replacing ‘x’ with ‘a’, e.g. (∀x)Px is true if and only if Pa, Pb, Pc, and so on are true. Likewise, an existentially quantified proposition (∃x)**P** is true iff for at least one name **a not** in **P** and at least one a-variant interpretation, it is the case that the **P(a/x)** is true (the formula that is the result of replacing ‘x’ with ‘a’).