Logique & Analyse **189–192** (2005), 189–207

CLASSICAL LOGIC WITH NON-REFERRING NAMES

RICHARD L. EPSTEIN[1]

The standard semantics for classical predicate logic require that for every name in the formal language there is some object in the universe to which it refers. In this paper I will show how we can lift that restriction in a simple way that allows for modeling various views of the role of non-referring names in reasoning.

A. *Logic for Nothing*

Lifting the restriction that names refer is often considered part of a program called *free logic*: ridding logic of existential assumptions that are built into its semantics. Along with no longer requiring names to refer, it is sometimes suggested that we should also lift the assumption that the universe of a model must contain at least one object. It is not for logic to assume that there is anything.

How then would we interpret 'Everything is a dog' if the universe is empty? Typically, 'All dogs bark' is taken to be equivalent to 'If there is anything that's a dog, then it barks,' where 'If ... then ...' is interpreted classically. The mathematicians' view that universal quantification should have no existential import is normally assumed as well: 'all' does not include 'and there exists'.[2] So 'Everything is a dog' should be interpreted as 'If there is anything, then it is a dog.' Thus 'Everything is a dog', and hence '$\forall x$ (x is a dog)', is true of the empty universe.

We don't need any complicated semantics for the logic of the empty universe. Every closed wff beginning with a universal quantifier is true; every closed wff beginning with an existential quantifier is false. The resulting

[1]Research for this paper was supported by a grant from the Advanced Reasoning Forum. I am grateful to Fred Kroon for discussions and many helpful suggestions in the preparation of this paper.

[2] See the discussion in Chapter V.B of *Epstein, 1994*.

formal system bears little resemblance to classical predicate logic. For example, $\forall x(x \neq x)$ is true in a model with empty universe, so $\neg\forall x(x \neq x)$ is no longer a tautology.

B. *Non-Referring Names in Classical Logic?*

There is a well-known way due to Bertrand Russell of formalizing descriptive names in classical predicate logic, including descriptive names that do not refer such as 'The cat that Richard L. Epstein likes.' Every apparently atomic wff in which a non-referring descriptive name appears is converted into a false proposition.[3]

With Russell's analysis we can never formalize a claim we believe is true that uses a non-referring name. But many say the following is true:

(1) Pegasus is a winged horse.

If the only semantic property of a name is whether it has a reference, and if so, what that reference is, it is hard to see how we can justify (1) as true. The most natural place to formalize reasoning with non-referring names is in a predicate logic in which further semantic values of language beyond truth-values and reference are taken into account, such as subject matter or referential content. Elsewhere I have shown how a large range of propositional logics take into account such semantic values.[4] In a subsequent work I will present predicate logics based on those logics in which the formalization of claims such as (1) might seem natural.

Nonetheless, we can develop semantics for non-referring names relative to the assumptions of classical predicate logic, viewing wffs with non-referring names as propositions, by taking as primitive whether a particular atomic proposition is true, just as we do in the pre-mathematical development of the usual semantics of classical predicate logic.[5] Such semantics will serve as a reference for predicate logics based on other semantic values and will also

[3] See Chapter VIII of *Epstein, 1994* for a presentation of Russell's analysis, along with a comparison to Peter Strawson's analysis of descriptive names.
W.V.O. Quine suggested that we can use the same method to eliminate all names, replacing, for example, 'Pegasus' with the predicate '– pegasizes'. That, he said, would allow us to formalize reasoning with atomic non-referring names, that is, ones that have no structure, *singular names*. But, as I have discussed in Chapter VIII.D of *Epstein, 1994*, Quine's suggestion miscontrues the role of names in reasoning.

[4] *Epstein, 1990*.

[5] See Chapter IV of *Epstein, 1994*.

allow a formalization of reasoning with partial functions, which is important in mathematics.

C. *Semantics for Classical Predicate Logic with Non-Referring Singular Names*

I'll first consider languages without function symbols and without the equality predicate. I'll extend the semantics to cover the equality predicate later in this section and leave languages with function symbols for Section F.

C.1. *Assignments of references and atomic predications*

Consider the semantics given for classical predicate logic before any mathematical abstractions are made.[6] We start with a non-empty universe and a complete set of assignments of references for variables. That is,

> There is at least one assignment of references. For every assignment of references σ, and every variable x, and every object of the universe, either σ assigns that object to x or there is an assignment τ that differs from σ only in that it assigns that object to x. If c is a name, then for every assignment of references σ assigns some object to c, and every other assignment of references assigns the same object to c.

We can extend our semantics by no longer requiring that each name have a reference. We need not assume that an assignment of references assigns any value to 'Pegasus'; indeed, since the only value it could assign in these semantics is a reference, it cannot assign a value. Thus, an assignment of references is a mapping from terms to objects of the universe satisfying:

(2)
i. For every variable x, $\sigma(x)$ is defined.
ii. For every name c, if $\sigma(c)$ is defined, then for every τ, $\tau(c)$ is defined and $\tau(c)$ is the same object as $\sigma(c)$.

For classical predicate logic we need not assume any particular metaphysics of truth in determining whether, say, 'Ralph is a dog' is true or whether $\sigma \vDash$ 'x is a dog' when $\sigma(x) =$ the brass lamp on my table (here "$\vDash$" is read as "validates"). We take the assignment of truth-values to atomic predications as primitive. Similarly, when we have non-referring names, we don't care how or why a claim such as (1) is true or false. All that matters is that (1) has a truth-value in the model.

[6] See Chapter IV of *Epstein, 1994*.

Now consider a binary predication:

(3) Pegasus is bigger than x.

If $\sigma(x)$ is Fred Kroon, how are we to determine the truth-value of (3)? Just as with (1), (3) is an atomic predication. It does not matter to the logician what truth-value is given to (3), so long as some truth-value is given. Thus, we have:

(4) For every assignment of references σ, whether $\sigma \vDash \mathrm{P}(t_1, ..., t_n)$ is taken as primitive.

Here $t_1, ..., t_n$ stand for any terms of the language.

This allows for the widest possible application of our logic, depending on how truth-values are assigned. It does not allow us to model the view that (1) has no truth-value; but that view is more naturally modeled within a many-valued logic than classical predicate logic.[7]

We need to make a restriction on (4). In classical predicate logic we assume that all predicates in our language are extensional: How we refer to an object in a predication does not matter for the truth-value of that predication. Given further semantic values, we could reason with non-extensional predicates and drop that requirement. But with the only semantic values available being truth-values and references, we are stuck with the assumption that predicates are extensional. We formulate a *condition on consistency and extensionality of predications* by modifying the one we that is (usually implicitly) adopted for classical predicate logic, noting that for any closed terms t and u, all we can take into account is whether $\sigma(t) = \tau(u)$ or whether t is u.[8]

(5) For all atomic wffs $Q(t_1, ..., t_n)$ and $Q(u_1, ..., u_n)$ and assignments σ and τ, if for each $i \leq n$, either $\sigma(t_i)$ is the same object as $\tau(u_i)$ or τ_i is a closed term and t_i is u_i, then

$$\sigma \vDash Q(t_1, ..., t_n) \text{ iff } \tau \vDash Q(u_1, ..., u_n).$$

[7] Some logicians think that using supervaluations makes a no-truth-value approach compatible with the semantics of classical logic, as described in *Bencivenga, 1986*.

[8] See *Epstein, 1994*, pp. 72–73, for the usual assumption.

C.2. *The quantifiers*

The interpretations I'll present here are what I consider the most reasonable in the context of giving the usual semantics for predicate logics.[9] Using these we'll see in Section E that we can model many other views of how to interpret the quantifiers.

C.2.1. *The universal quantifier*

For the universal quantifier we only have to note that an assignment of references does not take into account names that do not refer. Letting c range over names in the language, we have the *evaluation of the universal quantifier*:

(6) $\sigma \vDash \forall x$ A iff both

i. For every assignment of references τ that differs from σ at most in what it assigns as reference to x, $\tau \vDash \mathrm{A}(x)$.
ii. For every name symbol c, $\sigma \vDash \mathrm{A}(c)$.

Clause (ii) adds nothing when c refers; it is non-redundant only for non-referring names. Hence (6) *gives the same evaluation as used for models in which all names refer*. We are only making explicit a distinction between variables and names that was not needed when all names refer.

C.2.2. *The existential quantifier*

Suppose we take (1) to be true. Do we then conclude the following is true?

(7) $\exists x$ (x is a winged horse)

Opinion divides. Some say (7) is true *because* (1) is true. There is something of which 'x is a winged horse' is true, but not an "existent" thing. Others say (7) is false, because whatever role 'Pegasus' plays in our language, it is not a referring name, so it does not pick out something that exists.[10]

In the usual semantics for predicate logics, '$\exists$' is taken to mean 'there exists', and that is considered univocal. If we talk about different kinds of existence, then we should consider how atoms exist as different from dogs,

[9] Compare Chapter IV of *Epstein, 1994*.

[10] This view is usually modeled by assuming a second universe of non-existent objects. See, for example, *Bencivenga, 1986* and *Lambert, 1991*.

numbers as different from tables. We don't do that in establishing the foundations of predicate logic, though we can model such views within predicate logic by using predicates, such as '– is an abstract thing' or '– is a sensible object larger than 5 cm in diameter'.

So let's continue to take '$\exists$' to mean 'exists' in only one sense, which can be qualified, but doesn't also cover 'not exists'. Hence, (7) should come out false if the predicate 'winged horse' is not true of any thing, even if (1) is classified as true for whatever reason. So the evaluation of the existential quantifier will be the usual one:

$\sigma \models \exists x \mathrm{A}(x)$ iff for some τ that differs from σ at most in what it assigns x, $\tau \models \mathrm{A}(x)$

Thus, existential generalization, $\mathrm{A}(c) \rightarrow \exists x \mathrm{A}(x)$, is no longer valid. And the classical relation between $\forall$ and $\exists$ no longer holds, for the following can be false:

(8) $\neg \forall x \neg$ (x is a winged horse) $\rightarrow \exists x$ (x is a winged horse)

In a model with universe all living creatures, the antecedent can be true if (1) is classified as true, while the consequent would be false. We'll see in Section E, though, that we can model within these semantics the view that (8) is true.

C.3. *Summary of the semantics for languages without equality*

Since we need not alter our evaluations of the propositional connectives, the semantics for classical predicate logic with non-referring names amounts to making only the following changes to the standard semantics for classical predicate logic:

We allow for non-referring names, with assignments of references as at (2).
We adopt a condition on consistency and extensionality of predications (5).
We modify the evaluation of the universal quantifier as at (6).

We are only drawing distinctions we previously ignored: These semantics when used in a model in which only names refer give the same evaluation of wffs as the usual semantics.[11]

[11] So far as I can tell, these semantics are not equivalent to any others proposed as a free logic; compare *Bencivenga, 1986* and *Lambert, 1991*.

C.4. *Equality*

The equality predicate '$=$' is syncategorematic, a single interpretation for every model, relative to the universe of the model. So we should give an interpretation for it when we use non-referring names that is compatible with the interpretation when all names refer. We shall require that if $\sigma(t)$ and $\sigma(u)$ are both defined, then $\sigma \vDash t = u$ iff $\sigma(t)$ is the same object as $\sigma(u)$.

Now consider:

Pegasus $=$ Marilyn Monroe

This can't be true, because 'Pegasus' doesn't refer to any thing, while 'Marilyn Monroe' does. When one side of an equality refers and the other doesn't, the proposition is false. Even taking other semantic values into consideration, two names cannot refer in the same way if one refers to something and the other doesn't. Since for every assignment of references and every variable x, $\sigma(x)$ does refer to something, we'll also have that $\sigma \nvDash c = x$ if c does not refer.

Now consider:

Pegasus $=$ the horse beloved by Bellerophon

Both sides of the equality symbol are non-referring names. Any reason we have for saying this equality is true must be beyond the scope of classical predicate logic, for we have no other semantic value than reference to ascribe to names. (In semantics in which referential content is considered, for example, we might argue that this identity is true because the two sides point in the same way, regardless of their not having reference.) So just as with other atomic predications, we take as *primitive* whether $c = d$ is true when both c and d do not refer. But restrictions are needed.

First, consider:

Pegasus $=$ Pegasus

Whatever identity means, this must be true: That's just how we use identity. This view can be better defended when other semantic values are ascribed to names, saying, perhaps, that both sides of the equality symbol pick out in the same way, though they pick out nothing. Further, whatever value we give for $c = d$ we must give to $d = c$. And equalities should be transitive, too. We stray as little as possible from classical predicate logic.

There are some, like Russell, who hold that 'Pegasus $=$ Pegasus' should be false, since 'Pegasus' does not refer. We'll see in Section E how we can model that view within these semantics.

To summarize, letting σ, τ stand for any assignment of references, t, u, v for any terms and c, d for any names, we have the following.

Restrictions on the evaluation of the equality predicate

(9)
i. If both $\sigma(t)$ and $\sigma(u)$ are defined, then $\sigma \models t = u$ iff $\sigma(t)$ is the same object as $\sigma(u)$.
ii. If only one of $\sigma(t)$ and $\sigma(u)$ is defined, then $\sigma \nvDash t = u$.
iii. If both c and d do not refer, then for every σ and τ, $\sigma \models c = d$ iff $\tau \models c = d$.
iv. For all t, $\sigma \models t = t$.
v. For all t and u, $\sigma \models t = u$ iff $\sigma \models u = t$.
vi. For all t, u, and v, if $\sigma \models t = u$ and $\sigma \models u = v$, then $\sigma \models t = v$.

Condition (5) on consistency and extensionality of predications can now be more succinctly stated by taking into account the interpretation of the equality predicate:

(10) For all atomic wffs $Q(t_1, ..., t_n)$ and $Q(u_1, ..., u_n)$ and for any assignments σ and τ, if for each $i \leq n$, $\sigma \models t_i = u_i$, then

$$\sigma \models (Q(t_1, ..., t_n)) \text{ iff } \tau \models (Q(u_1, ..., u_n)).$$

Thus, the semantics for languages with the equality predicate require only (9) and (10) in addition to the previous semantics.

This completes the description of the semantics for classical predicate logic with identity and non-referring names.

In *Epstein, 2005* I present a mathematical abstraction of these semantics in which each predicate in a model is identified with its extension, which is taken to be a set, and the truth of atomic predications is determined by whether an element or sequence of elements of the universe is an element of that set.

D. *An Axiomatization*

I present here an axiomatization of *classical predicate logic with equality and non-referring names* for the language;

$$L(\neg, \rightarrow, \forall, \exists, =, P_0, P_1, ..., c_0, c_1, ...).$$

The axiomatization has to yield a collection of theorems of classical predicate logic, since every model of classical predicate logic with only referring names is a model here, too. Hence, every theorem (consequence) of classical predicate logic with non-referring names must also be a theorem (consequence) of classical predicate logic.

Propositional Axioms

The axiom schema of PC in L(¬, →), where A, B, C are replaced by predicate logic wffs and the universal closure is taken:

$\forall ... \neg \mathrm{A} \rightarrow (\mathrm{A} \rightarrow \mathrm{B})$
$\forall ... \mathrm{B} \rightarrow (\mathrm{A} \rightarrow \mathrm{B})$
$\forall ... (\mathrm{A} \rightarrow \mathrm{B}) \rightarrow ((\neg \mathrm{A} \rightarrow \mathrm{B}) \rightarrow \mathrm{B})$
$\forall ... (\mathrm{A} \rightarrow (\mathrm{B} \rightarrow \mathrm{C})) \rightarrow ((\mathrm{A} \rightarrow \mathrm{B}) \rightarrow (\mathrm{A} \rightarrow \mathrm{C}))$

Axioms governing $\forall$

$\forall ... (\forall x(\mathrm{A} \rightarrow \mathrm{B}) \rightarrow (\forall x \mathrm{A} \rightarrow \forall x \mathrm{B}))$ *distribution of* $\forall$
$\forall ... (\forall x \forall y \mathrm{A} \rightarrow \forall y \forall x \mathrm{A})$ *commutativity of* $\forall$

When x is not free in A, *superfluous quantification*
$\forall ... (\forall x \mathrm{A} \rightarrow \mathrm{A})$
$\forall ... (\mathrm{A} \rightarrow \forall x \mathrm{A})$

When term t is free for x in A, *universal instantiation*
$\forall ... (\forall x \mathrm{A}(x) \rightarrow \mathrm{A}(t/x))$

Axioms governing the relation between $\forall$ *and* $\exists$

$\forall ... (\exists x \mathrm{A} \rightarrow \neg \forall x \neg \mathrm{A})$
$\forall ... (\forall x \mathrm{A}(x) \rightarrow \exists x \mathrm{A}(x))$ *all implies exists*

Axioms for Equality

$\forall x(x = x)$

For every n-ary atomic predicate P,
$\forall ... (\bigwedge_i (t_i = u_i) \rightarrow (\mathrm{P}(t_1, ..., t_n) \rightarrow \mathrm{P}(u_1, ..., u_n))$

Axioms for non-referring names

$\forall ...(\forall y(\exists x(x = y) \to \neg B(y/x)) \to \neg\exists x B(x))$

$\forall ...((A(t/x) \wedge \exists x(x = t)) \to \exists x A(x))$ *existential generalization for referring names*

When x is not free in A, *superfluous quantification*

$\forall ...(\exists x A \to A)$

$\forall ...(A \to \exists x A)$

Rule *modus ponens* $\frac{A, A \to B}{B}$ where A and B are closed formulas

In *Epstein, 2005* I compare this axiomatization to one for classical predicate logic when all names are required to refer and establish the following, modifying the usual Henkin-style completeness proof for the case when all names refer.

Theorem In $L(\neg, \to, \forall, \exists, =, P_0, P_1, ..., c_0, c_1, ...)$ with non-referring names:

a. Every consistent collection of closed wffs in L has a countable model.

b. For any collection of closed wffs Σ in L, Σ is a complete and consistent theory iff there is a countable model $\mathcal{M}$ such that $\Sigma = \text{Th}(\mathcal{M})$.

c. For any model $\mathcal{M}$ of L, there is a countable model $\mathcal{M}*$ such that $\text{Th}(\mathcal{M}) = \text{Th}(\mathcal{M}*)$.

d. *Strong completeness* For any collection of closed wffs Γ and closed wff A, $\Gamma \vdash A$ iff $\Gamma \models A$.

e. *Compactness* For any collection of closed wffs Γ, Γ has a model iff every finite subset of Γ has a model.

E. *Examples of Formalization*

1. Pegasus is a winged horse.
   Therefore, something is a winged horse.

Analysis On the usual interpretation of the ordinary English in our models, the formalization of the example is:

Pegasus is a winged horse.
$\exists x$ (x is a winged horse)

The inference is invalid: In any model in which 'Pegasus' does not refer, the conclusion is false even if the antecedent is true. Existential generalization fails.

There are some, though, who say Example 1 is valid. This seems to me a remnant of the idea that the use of any name entails existence of a referent, which is what we set out to deny. There simply are no winged horses.

That view, though, becomes more respectable if we argue that there is a difference between 'there exists' and 'there is' (or 'something'). The former requires existence, and it is that which we have modeled with '$\exists$' in our system. But the latter does not. That is, the following is invalid:

Pegasus is a winged horse.
Therefore, there exists a winged horse.

But Example 1 and the following are valid:

Pegasus is a winged horse.
Therefore, there is a winged horse.

We can define within our system a *generous existence quantifier* to model that view of 'there is':

$$\exists_{\mathrm{G}} x \mathrm{A}(x) \equiv_{\mathrm{Def}} \neg \forall x \neg \mathrm{A}(x)$$

Then the following is valid:

Pegasus is a winged horse.
$\exists_{\mathrm{G}} x$ (x is a winged horse)

When x is the single variable free in A and c is a name, the following is valid:

$$\mathrm{A}(c/x) \vDash \exists_{\mathrm{G}} x \mathrm{A}(x)$$

2. Everything that is a horse is a mammal.
   Pegasus is a horse.
   Therefore, Pegasus is a mammal.

Analysis On the usual interpretation of the ordinary English, the formalization of Example 2 is:

$\forall x$ (x is a horse $\rightarrow$ x is a mammal)
Pegasus is a horse.
———————————
Pegasus is a mammal.

This is valid: It follows from universal instantiation and *modus ponens*.

Some say the example is not valid. We could have a model in which the universe is all animals that have ever lived, and then the premises are true even though 'Pegasus' does not refer, while the conclusion is false. On this view, 'everything' is interpreted as meaning 'Every existing thing': 'all' includes 'and there exists'. That is the interpretation of 'all' which is rejected in standard classical logic formalizations: 'All cats that are dogs are loyal' is counted as true. Mathematicians use 'all' without including 'and there exists'.

Nonetheless, we can model the view that 'all' includes 'and there exists' within this system by defining a *restricted universal quantifier*:

$\forall_{\mathrm{R}} x \mathrm{A}(x) \equiv_{\mathrm{Def}} \neg\exists x \neg \mathrm{A}(x)$

Then we would formalize Example 2 as:

$\forall_{\mathrm{R}} x$ (x is a horse $\rightarrow$ x is a mammal)
Pegasus is a horse.
———————————
Pegasus is a mammal.

This is invalid: the premises are true in the model described above, but the conclusion is false. Instantiation fails for the restricted universal quantifier.

3. Everything that is loved by Bellerophon is a winged horse.
   Pegasus is loved by Bellerophon.
   Therefore, something that is loved by Bellerophon is a winged horse.

Analysis On our interpretation of the ordinary English in our models, the formalization of Example 3 is:

$\forall x$ (x is loved by Bellerophon $\rightarrow$ x is a winged horse)
Pegasus is loved by Bellerophon.
———————————
$\exists x$ (x is loved by Bellerophon $\wedge$ x is a winged horse)

This is invalid: the conclusion could be false in a model if there are no winged horses, yet the premises could be true.

However, we can use the other ways to interpret the quantifiers described in the last two examples to formalize this example:

$\forall x$ (x is loved by Bellerophon $\rightarrow$ x is a winged horse) *valid*
Pegasus is loved by Bellerophon.
$\exists_{\mathrm{G}} x$ (x is loved by Bellerophon $\wedge$ x is a winged horse)

$\forall_{\mathrm{R}} x$ (x is loved by Bellerophon $\rightarrow$ x is a winged horse) *valid*
Pegasus is loved by Bellerophon.
$\exists x$ (x is loved by Bellerophon $\wedge$ x is a winged horse)

$\forall_{\mathrm{R}} x$ (x is loved by Bellerophon $\rightarrow$ x is a winged horse) *valid*
Pegasus is loved by Bellerophon.
$\exists_{\mathrm{G}} x$ (x is loved by Bellerophon $\wedge$ x is a winged horse)

4. Pegasus is Pegasus.

Analysis On our interpretation of the ordinary English, this is formalized as Pegasus = Pegasus. In all our models this is true: It's an instance of identity, $\forall x(x = x)$, which is valid.

Some say the example is false because Pegasus does not exist.[12] We can model that view by defining a *restricted equality*:

$x =_{\mathrm{R}} x \equiv_{\mathrm{Def}} \exists y(y = x \wedge x = x)$
where y is the least variable different from x

Then $\forall x(x =_{\mathrm{R}} x)$ will fail in any model in which there is a non-referring name. We can have 'Pegasus = Pegasus' is true, but 'Pegasus $=_{\mathrm{R}}$ Pegasus' is false.

Similarly, given any predicate $\mathrm{A}(x)$ with the single variable x free, we can define the *restriction of the predicate* A to be:

$\mathrm{A}_{\mathrm{R}}(x) \equiv_{\mathrm{Def}} \exists y(y = x \wedge \mathrm{A}(x))$
where y is the least variable that does not appear in $\mathrm{A}(x)$

Then $\mathrm{A}_{\mathrm{R}}(c/x)$ is false for any name c that does not refer.

[12] See the discussion in Chapter V.B of *Epstein, 1994*.

F. *Classical Predicate Logic with Names for Partial Functions*

F.1. *Partial functions in mathematics*

Mathematicians regularly use function names to create terms that name nothing. Such function names are meant to stand for partial functions, or as mathematicians say, functions whose domain is not the entire universe. For example, the name '$\tan$' stands for the tangent function in studies of the real numbers, and '$\tan(\pi/2)$' has no reference, nor does '$\tan(x)$' when x stands for $-3\pi/2$. The name '$-$' is used for the subtraction function on natural numbers, and '$5 - 7$' stands for no natural number.

Mathematicians try to avoid using compound names that do not refer by saying a function such as tangent is defined only for numbers other than $m\pi + n\pi/2$ for m any integer and n an odd integer; an expression such as '$\tan(\pi/2)$' is not a legitimate term. But to follow that line in our formalizations creates a serious problem. It is not trivial to determine for what values $\cot(\tan(\sqrt{x + \sin(y + \pi/3)}) + \cot(z))$ is defined. If to decide whether a concatenation of symbols of the formal language is a term we have to be able to decide an existence question, then the semantics become thoroughly enmeshed with the formation rules of the language. We would not be able to give an inductive definition of the formal language.

In this section I'll present a formal system for reasoning about partial functions in which compound terms need not refer.

F.2. *Semantics for partial functions*

The semantics for languages with names for partial functions is a modification of the semantics for languages with non-referring singular names.[13] Assignments of references are partial functions from terms to elements of the universe that satisfy:

(11)
i. For every variable x, $\sigma(x)$ is defined.
ii. For every closed term u, if $\sigma(u)$ is defined, then for every τ, $\tau(u)$ is defined and $\tau(u)$ is the same object as $\sigma(u)$.
iii. For every term t and every function name f in the language, if $\sigma(t)$ is not defined, then for every sequence of terms $t_1, ..., t, ..., t_n$, $\sigma(f(t_1, ..., t, ..., t_n))$ is not defined.

[13] See Chapter IX.E of *Epstein, 1994* for a survey of ways that have been proposed to reason using descriptive phrases such as 'the wife of –' as partial functions.

Condition (iii) reflects the usual practice that, for example, $\sin(\tan(\pi/2))$ is undefined because $\tan(\pi/2)$ is undefined.

The condition on the consistency and extensionality of predications (10) remains the same, since it was previously framed for any non-referring terms. But now we add a similar condition that *functions are extensional*:

(12) For all terms $f(t_1, ..., t_n)$ and $f(u_1, ..., u_n)$ and for any assignment σ, if for each $i \leq n$, $\sigma \vDash t_i = u_i$, then $\sigma \vDash f(t_1, ..., t_n) = f(u_1, ..., u_n)$.

However, we need something more:

(13) Assignments of references that agree on all the variables in two terms agree on on those terms.

This seems an essential part of what we mean by saying that applications of functions are extensional. Yet the restrictions (9) on the equality predicate plus (12) do not give us (13), for we could have a model satisfying both (9) and (12) in which c and d do not refer, $\sigma(x)$ is the same object as $\tau(x)$, yet $\sigma \vDash f(x, c) = d$ while $\tau \nvDash f(x, c) = d$. So we modify the restriction on the evaluation of the equality predicate (9) to read:

(14) (9.i–vi) except that (iii) is replaced by:

$\text{iii}_{\text{functions}}$. If for every variable x that appears in t or u, $\sigma(x)$ is the same object as $\tau(x)$, then $\sigma \vDash t = u$ iff $\tau \vDash t = u$.

The only other modification is to (6) to obtain the *evaluation of universal quantifiers for languages with non-referring names and partial functions*:

(15) $\sigma \vDash \forall x$ A iff

i. For every assignment of references τ that differs from σ at most in what it assigns as reference to x, $\tau \vDash$ A(x).

and

ii. For every τ and every term t free for x in A, $\tau \vDash$ A(t/x).

Clause (ii) requires that we survey not only all elements of the universe as we did when names refer, but all terms, too, along with all assignments to those. We are only drawing distinctions that we previously ignored. Except for allowing non-referring names and partial functions in our language, *there is nothing new in our semantics for classical predicate logic with non-referring names and partial functions*.

F.3. *Examples*

a.

> The extended real number system consists of the real number system to which two symbols, $+\infty$ and $-\infty$ have been adjoined, with the following properties:
>
> (a) If x is real, then $-\infty < x < +\infty$, and
> $x + \infty = +\infty, x - \infty = -\infty$, ...
>
> *Rudin*, 1964

> We can consider a *complex number* as having the form $a + bi$ where a and b are real numbers and i, which is called the *imaginary unit*, has the property that $i^2 = -1$. ...
>
> In performing operations with complex numbers we can proceed as in the algebra of real numbers, replacing i^2 by -1 when it occurs.
>
> 1. *Addition* $(a+bi)+(c+di) = a+bi+c+di = (a+c)+(b+d)i$
>
> *Spiegel*, 1964

Mathematicians often aren't clear whether they're using symbols such as '∞' and 'i' as names of things that exist or as nonreferring names, simply giving the rules for which atomic predications using them are true and which are false.

The main reason for introducing the symbols '$+\infty$' and '$-\infty$', however, is to have values for functions that would otherwise be undefined. For example, $\tan(\pi/2) = \infty$ and $\tan(3\pi/2) = -\infty$. If we read the new symbols as nonreferring names, we can model such reasoning with the semantics above. We can model reasoning with expressions such as '$\tan(\pi/2) = \tan(\pi + \pi/2)$' and '$\tan(\pi/2) \neq \tan(3\pi/2)$'.

b. In recursive function theory, mathematicians use a notion of equality for non-referring terms:

$f(x) \simeq g(x) \equiv f(x)$ and $g(x)$ are both defined and equal,
or both are undefined

All undefined terms are taken as equal, since undefined terms all arise in the same way: a calculation does not halt. We can model such reasoning with the semantics given above.

c. Here is a simple model that shows how complicated these semantics can become.

$\mathcal{U} = \{1, 2, 3, 4, 5\}$	
referring names:	none.
non-referring names:	a, b, c, d.
function symbol f	Interpret this as the partial function f, where $f(1) = 2$ and $f(3) = 3$.
equality	The evaluation is the least collection of pairs of assignments of references and equality wffs given the conditions of (14) plus, for all σ:

$\sigma \vDash a = b$	$\sigma \vDash f(a) = f(d)$	
$\sigma \nvDash a = c$	$\sigma \vDash f(b) = f(c)$	
(so by 9.vi, $\sigma \nvDash b = c$)	$\sigma \vDash f(c) = f(x)$	when $\sigma(x) = 2$
$\sigma \nvDash a = d$	$\sigma \vDash f(f(c)) = f(x)$	when $\sigma(x) = 4$
$\sigma \nvDash b = d$		
$\sigma \nvDash c = d$		

binary predicate symbol P	Interpreted as $\mathcal{P}$, where for every σ, $\sigma \vDash P(x, y)$ (i.e., P is interpreted as the universal function on $\mathcal{U}$) and for all σ:

$\sigma \nvDash P(a, y)$	when $\sigma(y) = 4$
$\sigma \nvDash P(x, c)$	when $\sigma(x) = 2$
$\sigma \nvDash P(f(x), c)$	when $\sigma(x) = 4$

d. Modify the previous model by first taking $\mathcal{A}$, an undecidable set of natural numbers . Writing 'f^n' for the iteration n times of f, set:

$$\sigma \vDash f^n(c) = f(x) \text{ iff } \sigma(x) = 4 \text{ for } n > 2 \text{ and } n \in \mathcal{A}$$

Even though the resulting model $\mathcal{M}$ has a finite universe and only finitely many names, functions, and predicates, the set of wffs true in $\mathcal{M}$ is not decidable. In particular, $\{n : \mathcal{M} \vDash \exists x(f^n(c) = f(x))\}$ is not decidable.

In contrast, for any model in which there are no partial functions and which has a finite universe and only finitely many names, functions, and predicates, the set of wffs valid in the model is decidable. Hence, we have shown the following, which is in contrast to classical predicate logic with referring names and total functions.[14]

Theorem Functions cannot in general be translated into predicates in classical predicate logic with non-referring names and partial functions.

We could ensure the translation of functions into predicates by allowing for partial predicates, saying, for example, that 'Pegasus is bigger than Juney' has no truth-value. But that is a major departure from classical logic which does not seem justified by the enjoinder to make logic free of existential assumptions (see the discussion below (4) above).

To my knowledge, no one proposes models as complicated as (d). Stipulations are usually made, such as that for all σ, for all non-referring terms t, u, $\sigma \nvDash t = u$, or as in (a) above, certain symbolic elements are added to the universe that determine the collection of equalities in a constructive manner. The evaluations of predicates are also usually simplified, for example, requiring $\sigma \nvDash P_i^n(t_1, ..., t_n)$ if $\sigma(t_i)$ is not defined for some i.

F.4. *An axiomatization*

To axiomatize *classical predicate logic with non-referring names and partial functions* we add just one axiom scheme to our previous list.

Axioms for functions

For every n-ary function symbol f,
$\forall ...(\bigwedge\!\!\!\bigwedge_i (t_i = u_i) \rightarrow (\mathrm{f}(t_1, ..., t_n) = \mathrm{f}(u_1, ..., u_n)).$

Since this axiom is true in all models in which all names refer and all functions are total, if $\Gamma \vdash$ A in this axiomatization, then A is a consequence of Γ in classical predicate logic.

The proof of strong completeness for this logic is presented in *Epstein, 2005*, where a mathematical abstraction of the semantics is also given.

Advanced Reasoning Forum
P.O. Box 635
Socorro, NM 87801
USA
E-mail: rle@AdvancedReasoningForum.org

[14] Compare Chapter VIII.E of *Epstein, 2005*.

REFERENCES

BENCIVENGA, Ermanno
1986 Free logics
Chapter 6 of Volume III of *Handbook of Philosophical Logic*, eds. D. Gabbay and F. Guenther, D. Reidel (Kluwer), pp. 373–426.

EPSTEIN, Richard L.
The Semantic Foundations of Logic:
1990 *Propositional Logics*
Kluwer. 2nd edition with corrections, Wadsworth, 2001.
1994 *Predicate Logic*
Oxford University Press. Reprinted 2001, Wadsworth.
2006 *Classical Mathematical Logic*
Princeton University Press.

LAMBERT, Karel
1991 *Philosophical Applications of Free Logic*
Oxford University Press.

RUDIN, Walter
1964 *Principles of Real Analysis*
McGraw-Hill, 2nd Edition.

SPIEGEL, Murray R.
1964 *Complex Variables*
Schaum's Outline Series, Mc-Graw Hill.