A note on identity and higher-order quantification.

Rafał Urbaniak
Centre for Logic and Philosophy of Science, Ghent University, Belgium
Chair of Logic, Methodology and Philosophy of Science, Gdańsk University, Poland
http://entiaetnomina.blogspot.com/
rfl.urbaniak@gmail.com

Received by Greg Restall
Published July 6, 2009
http://www.philosophy.unimelb.edu.au/ajl/2009

Abstract: It is a commonplace remark that the identity relation, even though not expressible in a first-order language without identity with classical set-theoretic semantics, can be defined in a language without identity, as soon as we admit second-order, set-theoretically interpreted quantifiers binding predicate variables that range over all subsets of the domain. However, there are fairly simple and intuitive higher-order languages with set-theoretic semantics (where the variables range over all subsets of the domain) in which the identity relation is not definable. The point is that the definability of identity in higher-order languages not only depends on what variables range over, but also is sensitive to how predication is construed. This paper is a follow-up to (Urbaniak 2006), where it has been proven that no actual axiomatization of Leśniewski's Ontology determines the standard semantics for the epsilon connective.

1 INTRODUCTION

Say we have a first-order language L_1 with a countable assembly of predicates, which contains no other extralogical symbols, and whose logical symbols, besides brackets and a countable set of individual variables, are the classical Boolean connectives and the classical quantifiers binding individual variables. Most importantly, the identity symbol is not among the logical symbols of that language. Consider its standard set-theoretic semantics. For any set Γ of formulas of L_1, let $\mathbb{M}(\Gamma)$ be the set of all those models which model all formulas from Γ. The following is a well-known fact:[1]

FACT 1 For any two-place predicate R of L_1, for any set Γ of formulas from L_1, there is a model M in $\mathbb{M}(\Gamma)$ such that the interpretation of R in M (R^M) is not the identity relation in M (i.e. $R^M \neq \{\langle x, x\rangle \mid x \in M\}$). □

[1] For a proof, see for example (Manzano 1996).

In other words, this means that in L_1 we can't specify conditions that would define R to play the semantic role of the identity symbol.

What happens when we extend the language to L_2 by adding predicate variables and quantifiers binding them to L_1's logical vocabulary (but still, not introducing the identity symbol as a logical primitive)? Well, if we interpret second-order quantifiers as ranging over all subsets of the domain,[2] we have another well-known fact:

FACT 2 The relation of identity is definable by means of a single formula. That is, if we take:

$$\forall_{x,y}[R(x,y) \equiv \forall_P(P(x) \equiv P(y))] \tag{1}$$

where R is a newly defined two-place predicate of L_2 and P is a predicate variable of L_2, then (1) is satisfied in a second-order model if and only if the interpretation of R in that model is the identity relation in its domain.[3] □

In a while, we'll move on to a higher-order language which doesn't contain a separate category of singular terms. To give the taste of how we can still speak of identity of individuals in such a language, we'll introduce a relation between predicates that can go proxy for such an identity relation. Extend the language of L_2 to $L_2^=$ by adding an identity symbol that connects predicate symbols, thus forming a formula '$P = Q$'. On the intended interpretation '$P = Q$' is true if and only if the referent of both P and Q is one and the same singleton. This identity relation is expressible already in L_2. Given that we define identity between individuals as in (1) and understand $\exists!$ as 'there is exactly one',[4] we can just say:

$$P = Q \equiv \exists!_x P(x) \wedge \exists!_x Q(x) \wedge \forall_x(P(x) \equiv Q(x))] \tag{2}$$

FACT 3 The relation of identity between predicates (as explained above) is definable by means of a single formula of L_2, namely (2). □

Prima facie, one might draw the moral that in order to ensure that the identity relations mentioned above are definable, it is enough to introduce second-order quantification. As it will turn out, there are languages in which quantification ranging over all subsets of the domain is available, but in which the identity relation is not definable, because predication functions differently.

2 QUANTIFIED NAMING LOGIC (QNL)

Suppose that instead of accepting the standard division between singular terms and predicate symbols we rather take the quasi-Aristotelian approach, found for instance in the language of Leśniewski's Ontology.[5] That is, we construct

[2] As opposed, for instance, to interpreting them as ranging over all finite or over all definable subsets of the domain.

[3] The proof is fairly simple, but the reader can consult (Manzano 1996) for details.

[4] Note that expressing '$\exists!$' requires identity, so the quantifier is not expressible in L_1.

[5] See (Urbaniak 2008) for more details and an extensive bibliography concerning Leśniewski's systems.

a language L_{ε} which contains only one category of variables, called *name variables*: $a_1, a_2, a_3, a_4 \ldots$ (abbreviated by $a, b, c, d, \ldots$). The set of all name variables is called $\mathbb{V}$. We keep the Boolean connectives and brackets, but we admit only one type of quantifiers: those binding name variables. Most importantly, we also introduce a new constant: ε. This new constant is used to form atomic formulas, so that the formation rules are as follows:

- If α and β are name variables, then $(\alpha\, \varepsilon\, \beta)$ is a well-formed formula.
- If ϕ and ψ are well-formed formulas, so are:

$$(\neg\phi), (\phi \wedge \psi), (\phi \vee \psi), (\phi \rightarrow \psi), (\phi \equiv \psi)$$

- If ϕ is a well-formed formula, and α a name variable, $\forall_\alpha(\phi)$ and $\exists_\alpha(\phi)$ are well-formed formulas.
- Nothing else is a well-formed formula.

The intended interpretation is that name variables represent names, which can be either empty or singular or plural, and ε expresses the 'is' or 'is one of' of predication. Hence, we read '$a\, \varepsilon\, b$' as 'a is one of b's', or simply as 'a is b'. The Boolean operators work in the standard manner and the quantifiers range over all subsets of the domain. There are quite a few ways one can give semantics for L_{ε}. We'll consider only two of them.

An S-model is a structure $M = \langle D, V\rangle$, where D is a non-empty domain of objects and V is a total function that maps the name variables into the powerset of D, 2^D. If we note 'A is a singleton' as '$\mathtt{Sing}(A)$', the satisfaction conditions are as follows:

$$\begin{aligned}
\langle D, V\rangle \models \alpha\, \varepsilon\, \beta \quad &\text{iff} \quad \mathtt{Sing}(V(\alpha)) \wedge V(\alpha) \subseteq V(\beta) \\
\langle D, V\rangle \models \phi \wedge \psi \quad &\text{iff} \quad \langle D, V\rangle \models \phi \text{ and } \langle D, V\rangle \models \psi \\
\langle D, V\rangle \models \neg\phi \quad &\text{iff} \quad \langle D, V\rangle \not\models \phi \\
\langle D, V\rangle \models \exists_\alpha \phi \quad &\text{iff} \quad \langle D, V'\rangle \models \phi \\
&\qquad \text{for some } V' \text{ that differs from } V \text{ at most at } \alpha.
\end{aligned} \tag{3}$$

Satisfaction conditions for other Boolean clauses and for the universal quantifier can be defined in terms of the satisfaction clauses defined above (as usual, the quantifiers are interdefinable, and the definitions of other Boolean quantifiers are standard). To avoid ambiguities, I sometimes use '$\models_s$' to denote this satisfaction relation.

An N-model is a structure $M = \langle D, R\rangle$, where D is a non-empty domain of objects and $R \subseteq \mathbb{V} \times D$ is a "naming relation". If we have two naming relations R and R', we say that they differ at most on a variable α if and only if:

$$\{\langle \beta, y\rangle \mid \beta \neq \alpha \wedge \langle \beta, y\rangle \in R\} = \{\langle \beta, y\rangle \mid \beta \neq \alpha \wedge \langle \beta, y\rangle \in R'\} \tag{4}$$

The satisfaction conditions differ from the ones given in (3) only for atomic formulas. All other clauses are, *mutatis mutandis*, the same.

$$\langle D, R\rangle \models \alpha\, \varepsilon\, \beta \quad \text{iff} \quad \exists!_x R(\alpha, x) \wedge \forall_y [R(\alpha, y) \rightarrow R(\beta, y)] \tag{5}$$

I sometimes use '$\models_n$' to denote this satisfaction relation. There is a slight difference between S-models and N-models. In an S-models all names "refer to" sets, so that each name refers to exactly one subset of a domain. In an N-model, R is rather a direct, and possibly multiple reference relation between variables and objects themselves. A variable α can:

- Not to refer to anything: $\forall_{y \in D} \neg R(\alpha, y)$
- Refer to exactly one object: $\exists!_{x \in D} R(\alpha, x)$
- Refer to more than one object: $\exists x, y \in D[x \neq y \wedge R(\alpha, x) \wedge R(\alpha, y)]$

Despite this difference, there is an obvious correspondence between S-models and N-models that preserves satisfaction.

Some observations are due. First, even though the language of QNL is a sublanguage of the language of Leśniewski's Ontology (which contains variables and quantifiers for other syntactic categories as well), QNL is not a subsystem of Leśniewski's Ontology, because in set-theoretic semantics for Ontology it is not required that the domain be non-empty. This could be avoided, if we allowed for empty QNL models. Since, however, we are rather interested in comparing QNL and second-order logic which does embrace the non-emptiness assumption, this won't be done in this paper.

Second, say we have a language of monadic second-order logic whose all predicate variables are $P_{a_1}, P_{a_2}, P_{a_3}, \ldots$. Map (1-1) each name variable a_i to a predicate P_{a_i}. There is a translation from the language of QNL into this language that preserves satisfaction (*modulo* an obvious isomorphism of QNL models and monadic second-order models). The translation t goes as follows:

$$\begin{aligned} t(a\, \varepsilon\, b) &= \exists!_x P_a(x) \wedge \forall_x (P_a(x) \rightarrow P_b(x)) \\ t(\phi \wedge \psi) &= t(\phi) \wedge t(\psi) \\ t(\neg\phi) &= \neg t(\phi) \\ t(\exists_a \phi) &= \exists_{P_a} t(\phi) \end{aligned} \tag{6}$$

Third, an analogous translation from the language of second-order monadic predicate logic which goes the other way round is available. Order the name variables into two sequences:

$$a_1, a_2, a_3, \ldots$$
$$b_1, b_2, b_3, \ldots$$

Map each predicate P_i to a_i, and each individual variable x_i to b_i. Then the translation f can be defined:

$$\begin{aligned} f(P_i(x_j)) &= b_j \varepsilon a_i \\ f(\phi \wedge \psi) &= f(\phi) \wedge f(\psi) \\ f(\neg\phi) &= \neg f(\phi) \\ f(\exists_{x_i}\phi) &= \exists_{b_i}(b_i \varepsilon b_i \wedge f(\phi)) \end{aligned} \tag{7}$$

This indicates that QNL isn't that far from second-order logic – in fact, given that its semantics involves quantification over all subsets of a domain (or an analogous device of quantification over naming relations), it can be considered a higher-order logic.

Coming back to the main issue, the identity relation can be defined in L_ε by:

$$a = b \equiv a \varepsilon b \wedge b \varepsilon a \tag{8}$$

Indeed, (8) defines identity, because:

- $a \varepsilon b \wedge b \varepsilon a$ can be true in an S-model only if both $V(a)$ and $V(b)$ are both singletons contained in each other, which can be the case only if they are both one and the same singleton (in a sense, the identity of singletons "goes proxy" for the real identity of objects).
- $a \varepsilon b \wedge b \varepsilon a$ can be true in an N-model only if a and b refer directly to one and the same object, and there are no other objects to which a and b refer.

3 UNDEFINABILITY ISSUES

So identity can be defined both in second-order logic and in QNL. In QNL, however, this fact depends not only on the behavior of quantifiers, but also on the meaning of the epsilon. Hence, the following question arises: can we characterize the semantic behavior of this copula syntactically, just like we can define identity in second-order logic? In other words, what happens if we replace ε in L_ε with f, a symbol that plays the same syntactic role as ε, thus obtaining a language L_f — can we give a set of formulas which makes f play the semantic role of ε? The answer is negative. Let's make the question more specific first.

An F_s-model of L_f is a tuple $\langle D, V, e\rangle$, where D is a non-empty set of objects, V is as it was in S-models, and e is a function that maps f into a relation between subsets of D: $f \subseteq 2^D \times 2^D$. The satisfaction clause for the atomic case is:

$$\langle D, V, e\rangle \models \alpha f \beta \quad \text{iff} \quad \langle V(\alpha), V(\beta)\rangle \in e(f) \tag{9}$$

The other clauses are just as in (3). This satisfaction relation will be sometimes denoted by '$\models_{fs}$'.

An F_n-model of L_f is a tuple $\langle D, R, e\rangle$, where D is a non-empty set of objects, R is as it was in N-models, and e is as it was in the definition of F_s-models. The satisfaction clause for atomic formulas is:

$$\langle D, R, e\rangle \models \alpha f \beta \quad \text{iff} \quad \langle \{x \in D \mid R(\alpha, x)\}, \{x \in D \mid R(\beta, x)\}\rangle \in e(f) \qquad (10)$$

The rest of the conditions remains the same. For this satisfaction relation I sometimes write '$\models_{fn}$'.

Let's say that a model which contains a function e is standard if and only if $e(f)$ is a relation that holds between two subsets A, B of D if and only if A is a singleton which is a subset of B (that is, if f is interpreted as the epsilon). We have the following fact:

FACT 4 No set of L_f-formulas Γ is such that for any F_s-model $\langle D, V, e\rangle$:[6]

$$\langle D, V, e\rangle \models_{fs} \Gamma \text{ iff } \langle D, V, e\rangle \text{ is standard.} \qquad (11)$$

Argument. For any F_s-model $M = \langle D, V, e\rangle$ where $e(f)$ is the standard interpretation of ε, take an object $\perp \notin D$ and let $M' = \langle D \cup \{\perp\}, V', e'\rangle$ where for any α:

$$V'(\alpha) = V(\alpha) \cup \{\perp\}$$

and $e'(f)$ is defined by saying that for any two subsets A, B of $D \cup \{\perp\}$, $\langle A, B\rangle \in e'(f)$ iff (i) $A \subseteq B$, and (ii) there are exactly two objects in A. By induction on the length of a formula, M and M' agree on all formulas, and hence for any set of formulas Γ, $M \models_{fs} \Gamma$ iff $M' \models_{fs} \Gamma$.

To see why this is the case, consider the satisfaction condition for atomic sentences. Since the interpretation of f in M is standard, we know that M models $\alpha f \beta$ iff $V(\alpha)$ is a singleton which is a subset of $V(\alpha)$. But this can be the case if and only if both (i) $V'(\alpha)$ contains exactly two elements: $\perp$ and the object that originally was in $V(\alpha)$, and (ii) $V'(\alpha)$ is a subset of $V'(\beta)$. The induction steps for Boolean connectives are simple.

The claim holds also for quantified statements. For suppose:

$$\langle D, V, e\rangle \models \exists_\alpha \phi(\alpha)$$

Then, there is a V_i that differs from V at most on α, such that $\langle D, V_i, e\rangle \models \phi(\alpha)$. Construct a model $\langle D \cup \{\perp\}, V_i', e'\rangle$, where e' is as already defined and for any α, $V_i'(\alpha) = V_i(\alpha) \cup \{\perp\}$. Clearly, $\langle D \cup \{\perp\}, V_i', e'\rangle \models \phi(\alpha)$ and hence our $\langle D \cup \{\perp\}, V', e'\rangle$ models $\exists_\alpha \phi(\alpha)$. Implication in the opposite direction holds for similar reasons.

Now suppose Γ satisfies (11) for any F_s-model. Clearly, there are standard F_s-models. Take one of those, call it M. Generate M' according to the instructions give above. Γ will also be satisfied in M', but M' isn't standard, which means that Γ doesn't define f to be the epsilon. □

[6] Modeling a set of formulas is just modeling all its members.

This also means that the identity between singletons is not definable in L_f within F_s semantics.

COROLLARY 5 *No* L_f *formula* ϕ *containing two free variables* α *and* β *is such that for any* F_s*-model* $\langle D, V, e\rangle$*:*

$$\langle D, V, e\rangle \models \phi \textit{ iff } \mathtt{Sing}(V(\alpha)) \wedge \mathtt{Sing}(V(\beta)) \wedge V(\alpha) = V(\beta)$$

Argument. Use the same construction as in the proof of fact 4 to show that there is a non-standard model $\langle D', V', e'\rangle$ where $\phi(\alpha, \beta)$ is satisfied, even though it is not the case that $V'(\alpha)$ and $V'(\beta)$ are one and the same singleton. □

The following corollaries have analogous proofs:

COROLLARY 6 *No set of* L_f*-formulas* Γ *is such that for any* F_n*-model* $\langle D, R, e\rangle$*:*

$$\langle D, R, e\rangle \models_{fn} \Gamma \textit{ iff } \langle D, R, e\rangle \textit{ is standard.} \tag{12}$$

COROLLARY 7 *No* L_f *formula* ϕ *containing two free variables* α *and* β *is such that for any* F_n*-model* $\langle D, R, e\rangle$*:*

$$\langle D, R, e\rangle \models \phi(\alpha, \beta) \textit{ iff } \exists!_x R(\alpha, x) \wedge \exists!_x R(\beta, x) \wedge \forall_x [R(\alpha, x) \equiv R(\beta, x)]$$

This last corollary also means that the identity between objects is not definable in L_f with respect to F_n-models.

Interestingly, F_s-models can be viewed as higher-order standard model (we can quantify over all subsets of the domain) and F_n-models, as far as expressive power is concerned, can also be treated as such, given that there's an obvious 1-1 correspondence between F_n- and F_s- models.

These considerations may suggest that a key role in providing standard second-order languages with extra expressive power is played not only by quantification over all subsets of the domain, but also by the distinction between singular terms and predicate symbols. A classical second-order language allows us to ensure that an expression refers to exactly one object: one just needs to use a singular variable. L_f is devoid of this feature and this might make quite a difference. This diagnosis, however, is not quite adequate. Introducing singular terms doesn't solve the problem.

Suppose we extend L_f to L_f^i by adding quasi-individual variables, $i_1, i_2, i_3, \ldots$ (unofficially: $i, j, k, \ldots$). An F_s^i-model is an F_s-model with the valuation function V extended to map individual variables into the set of all *singletons* of objects from the domain (hence '*quasi*-singular'). An F_n^i-model is an F_n-model where R is extended to relate individual variables to elements of D, so that for any individual variable ι, $\exists!_{x \in D} R(\iota, x)$.

Notice that the identity between individual variables cannot be defined in the way that *prima facie* might seem plausible, i.e. by:

$$\mathfrak{i} = \mathfrak{j} \equiv \forall_{\mathfrak{a}}[\mathfrak{i}f\mathfrak{a} \equiv \mathfrak{j}f\mathfrak{a}]$$

because if $e(f)$ is the empty relation, then for any interpretation of $\mathfrak{i}$ and $\mathfrak{j}$ it will be the case $\forall_{\mathfrak{a}}[\mathfrak{i}f\mathfrak{a} \equiv \mathfrak{j}f\mathfrak{a}]$.

The construction method from the proof of fact 4 no longer works for L_f^i, because it does't generally preserve satisfaction of formulas in which individual variables occur. There is, however, a brute-force way of modifying the method so that it yields non-standard F_s^i-models (*mutatis mutandis*, F_n^i-models) that agree with (respective) standard models on all formulas. Say we have a standard F_s^i-model $M = \langle D, V, e\rangle$ and an object $\perp \notin D$. Let M' be $\langle D \cup \{\perp\}, V', e'\rangle$ such that:

- If α is a name variable, then $V'(\alpha) = V(\alpha) \cup \{\perp\}$.
- If ι is an individual variable, then $V'(\iota) = V(\iota)$.
- For any $A, B \in 2^D$, $\langle A, B\rangle \in e'(f)$ iff (i) $A \setminus \{\perp\}$ is a singleton, and (ii) $A \setminus \{\perp\} \subseteq B \setminus \{\perp\}$. That is, $\langle A, B\rangle \in e'(f)$ iff $\langle A \setminus \{\perp\}, B \setminus \{\perp\}\rangle \in e(f)$.

M' is a non-standard model that agrees with M on all formulas. This means that:

FACT 8 Neither identity nor the epsilon is definable in L_f^i.

This indicates that a more plausible diagnosis would be that the key difference between L_f and standard second-order languages, a difference that underlies the undefinability of identity in L_f, is that instead of taking the predication relation between a singular term and a predicate to be primitive, we take as our primitive the relation that holds between a copula and its two arguments.

References

MANZANO, M. (1996). *Extensions of First Order Logic*. Cambridge University Press.

URBANIAK, R. (2006). Some non-standard interpretations of the axiomatic basis of Lesniewski's Ontology. *The Australasian Journal of Logic*, 4:13–46.

URBANIAK, R. (2008). *Leśniewski's Systems of Logic and Mereology; History and Re-evaluation*. PhD thesis, University of Calgary. Available at https://dspace.ucalgary.ca/handle/1880/46697.

The *Australasian Journal of Logic* (ISSN 1448-5052) disseminates articles that significantly advance the study of logic, in its mathematical, philosophical or computational guises. The scope of the journal includes all areas of logic, both pure and applied to topics in philosophy, mathematics, computation, linguistics and the other sciences.

Articles appearing in the journal have been carefully and critically refereed under the responsibility of members of the Editorial Board. Only papers judged to be both significant and excellent are accepted for publication.

The journal is freely available at the journal website at

`http://www.philosophy.unimelb.edu.au/ajl/`.

All issues of the journal are archived electronically at the journal website.

Subscriptions Individuals may subscribe to the journal by sending an email, including a full name, an institutional affiliation and an email address to the managing editor at ajl-editors@unimelb.edu.au. Subscribers will receive email abstracts of accepted papers to an address of their choice. For institutional subscription, please email the managing editor at ajl-editors@unimelb.edu.au.

Complete published papers may be downloaded at the journal's website at `http://www.philosophy.unimelb.edu.au/ajl/`. The journal currently publishes in pdf format.

Submission The journal accepts submissions of papers electronically. To submit an article for publication, send the LaTeX source of a submission to a member of the editorial board. For a current list of the editorial board, consult the website.