

# A Conceptualist Interpretation of Leśniewski's Ontology

Nino B. Cocchiarella

## Abstract

A first-order formulation of Leśniewski's ontology is formulated and shown to be interpretable within a free first-order logic of identity extended to include nominal quantification over proper and common-name concepts. The latter theory is then shown to be interpretable in monadic second-order predicate logic, which shows that the first-order part of Leśniewski's ontology is decidable.

One of the important applications of Leśniewski's system of ontology, sometimes also called *the logic of (proper and common) names*<sup>1</sup>, has been as a logistic framework that can be used in the reconstruction of medieval terminist logic.<sup>2</sup> This is especially so because the basic relation of Leśniewski's system, singular inclusion, amounts to a version of the two-name theory of the copula.<sup>3</sup> An alternative reconstruction of medieval terminist logic can also be given within the logistic framework of conceptual realism, however.<sup>4</sup> This system is preferable in part because, unlike Leśniewski's ontology, it is not committed to an extensional framework, which is important in the logical analysis of the tensed and modal modification (ampliation) of the terms (names) of medieval logic.<sup>5</sup> It is also possible, as we show below, to reduce or

---

<sup>1</sup>See, e.g. [Lejewski 1958], p. 152, [Slupecki 1955] and [Iwanuś 1973].

<sup>2</sup>See, e.g., [Henry 1972].

<sup>3</sup>Singular inclusion, as represented by  $\varepsilon$  in the formula ' $a \varepsilon b$ ', read as ' $a$  is  $b$ ', was the only undefined constant of Leśniewski's original system of ontology. The system could also be based either on partial, weak or strong inclusion as the only primitive as well. See [Lejewski 1958]. pp. 154-156.

<sup>4</sup>See [Cocchiarella 2001]. For an account of conceptual realism as a formal ontology see [Cocchiarella 1996].

<sup>5</sup>One might, of course, add tense and modal operators to Leśniewski's ontology, even though he himself was against such a move.

reconstruct Leśniewski's ontology within the logistic framework of conceptual realism.

## 1 Leśniewski's Ontology as a First-Order Theory

There are three main parts to Leśniewski's general logistic framework. These are (1) a version of the simple theory of types, which he called the system of semantic categories and which includes protothetic, (2) the system of ontology as based on the relation of singular inclusion, and (3) mereology as based on the relation of part to whole.<sup>6</sup> We will not deal with mereology here at all. Protothetic, which is a quantificational logic over propositions and  $n$ -ary truth functions, for each positive integer  $n$ , is an interesting, but redundant part of the theory of types insofar as the logic is functionally complete, i.e., insofar as every  $n$ -ary truth function can be represented in the logic.<sup>7</sup> Aside from mereology, in other words, Leśniewski's logistic framework is essentially equivalent to the theory of simple types plus the system of ontology.<sup>8</sup> For our purposes here, however, even the theory of simple types can be ignored. Instead, we will represent Leśniewski's system of ontology as an applied first-order logic with  $\varepsilon$  as a two-place predicate constant representing the relation of singular inclusion as its only primitive descriptive constant. This is sufficient, we believe, because the logistic framework of conceptual realism is equivalent to the simple theory of types, and hence to the nonelementary part of Leśniewski's system of ontology; and also because, as we show here, a representation, or reduction, of the elementary part of Leśniewski's ontology can be given within a certain extension of the identity logic of the first-order part of the framework of conceptual realism.

It should be noted, however, that Leśniewski's theory of definitions for constant names and name-forming functors does not always satisfy the conditions for noncreativity that definitions are required to satisfy in standard first-order theories. For that reason we do not include such definitions in

---

<sup>6</sup>See [Slupecki 1955] for a description of (1) and (2).

<sup>7</sup>There are only a finite number of  $n$ -ary truth-functions for each positive integer  $n$ , and hence quantification over these truth-functions is equivalent to a conjunction or disjunction of  $n$  many conjuncts or disjuncts in which, ultimately, such quantification does not occur. See [Church 1956], §28.

<sup>8</sup>See [Slupecki 1955] for justification of this claim.

the first-order theory described below. There is a certain extension of this first-order theory, however, for which the noncreativity conditions of these definitions can be proved.<sup>9</sup> The axioms of this extension are also reducible, it turns out, to the part of the conceptualist logic of identity described below in section two. In other words, Leśniewski’s system of “elementary ontology,” as it sometimes called, is reducible to this conceptualist logic.<sup>10</sup> We will briefly sketch the proof for this claim in section four.

The so-called “nominal” (or “name”) variables  $a, b, c, d$  (with or without numerical subscripts) of Leśniewski’s system of ontology, for which the symbolic counterparts of shared, unshared or fictitious names, proper or common, can be substituted, will be taken to be the individual variables of our first-order version of the system. That is, the symbolic names that are the substituends of these variables (in particular applications of the ontology) are assumed to represent common names (common count nouns), including the ultimate superordinate common name ‘thing’, as well as proper names. The symbolic counterparts of fictitious names—i.e. vacuous names, whether proper or common, such as ‘Pegasus’ and ‘unicorn’—are also assumed to be legitimate substituends of the nominal variables. The symbolic counterparts of complex common names, such as ‘thing that is red’, are also names in the sense intended, which is an indirect way of including adjectives as names. For the formula ‘ $a \varepsilon b$ ’ to be true, however, the name ‘ $a$ ’ must denote and be unshared, i.e. ‘ $a$ ’ must then name exactly one thing.

The identity sign, ‘=’, which is used to represent the so-called “strong identity” of ontology, is not a primitive symbol of the system, but is defined instead as follows<sup>11</sup>:

**Definition 1**  $a = b \leftrightarrow a \varepsilon b \wedge b \varepsilon a.$

Note that for ‘ $a = b$ ’ to be true, both ‘ $a$ ’ and ‘ $b$ ’ must be unshared names of the same thing. This means that when ‘ $a$ ’ is a shared or vacuous name,

---

<sup>9</sup>See [Iwanuś 1973].

<sup>10</sup>We should perhaps emphasize that, as described in [Iwanuś 1973], there are no inference rules for the system of elementary ontology,  $EO$ , other than those for standard first-order logic without identity. That is, Leśniewski’s rules for definitions are eliminable in  $EO$ .

<sup>11</sup>We use  $\rightarrow, \neg, \wedge, \vee, \leftrightarrow$  as the (material) conditional, negation, conjunction, disjunction, and biconditional signs, and  $\forall,$  and  $\exists$  as the universal and particular quantifiers. For convenience of logical details, we assume that  $\rightarrow, \neg,$  and  $\forall$  are the only primitive logical constants, and that the others are defined in terms of these in the standard way.

‘ $a = a$ ’ will be false, which means that we do not have ‘ $(\forall a)(a = a)$ ’ as a valid thesis of the system. With weak identity, however, the situation is different. Weak identity, sometimes also called “equality”, is mutual subordination (or inclusion), which is defined as follows<sup>12</sup>:

**Definition 2**  $a \circ b \leftrightarrow (\forall c)(c \varepsilon a \leftrightarrow c \varepsilon b)$ .

Unlike ‘ $a = a$ ’, the strong identity of  $a$  with itself, the weak identity of  $a$  is unproblematic, i.e. ‘ $(\forall a)(a \circ a)$ ’ is a valid thesis. Indeed, it might well be preferable to use ‘ $\circ$ ’ for strong identity and ‘ $=$ ’ for weak identity instead, because then we would not have the appearance of conflict with the logic of identity (which is reflexive) when in fact there is no real conflict.<sup>13</sup>

In taking the symbolic counterparts of names, both proper and common, vacuous or otherwise, as substituends of the bound individual variables, we do not mean that quantification is to be interpreted substitutionally. That is, even though

$$\varphi(c/a) \rightarrow (\exists a)\varphi(a)$$

is a valid thesis (and in fact is the contrapositive of a logical axiom<sup>14</sup>) regardless whether  $c$  is a shared, unshared or fictitious name, this does not mean that quantifiers indexed by nominal variables must be interpreted substitutionally. Some authors, however, while maintaining that quantification in Leśniewski’s ontology should not be interpreted substitutionally, go on to claim that it also cannot be interpreted referentially. This is so, according to these authors, because there are more names than there are objects (or so they claim).<sup>15</sup> The implicit assumption, apparently, is that every object (including every grain of sand and every microparticle of the universe)

<sup>12</sup>See, e.g., [Küing & Canty 1970], p. 166.

<sup>13</sup>In [Iwanuś 1973], the identity sign, ‘ $=$ ’, is used in the sense we suggest, i.e. as co-extensivity with respect to  $\varepsilon$ .

<sup>14</sup>As logical axioms we assume all tautologous formulas as well as all instances of the following schemas:

$$\begin{aligned} &(\forall a)\varphi(a) \rightarrow \varphi(c/a), \\ &(\forall a)[\varphi \rightarrow \psi] \rightarrow [(\forall a)\varphi \rightarrow (\forall a)\psi], \\ &[\chi \rightarrow (\forall a)\chi], \end{aligned}$$

where  $a$  is not free in  $\chi$ , and  $c$  is free for  $a$  in  $\varphi$ . Modus ponens and universal generalization will suffice as inference rules

<sup>15</sup>See [Küing & Canty 1970], pp.169f.

has a unique unshared name, and that in addition to these there are many shared and fictitious names as well. The interpretation we give below in our conceptualist reduction of ontology does not depend on such a dubious assumption, however; nor does it assume that quantification with respect to nominal variables must be substitutional.

Finally, the only nonlogical axiom of ontology (i.e. the only axiom in addition to the logical axioms and inference rules of first-order predicate logic *without* identity) is the following:

$$(\forall a)(\forall b)[a \varepsilon b \leftrightarrow (\exists c)c \varepsilon a \wedge (\forall c)(c \varepsilon a \rightarrow c \varepsilon b) \wedge (\forall c)(\forall d)(c \varepsilon a \wedge d \varepsilon a \rightarrow c \varepsilon d)].$$

We will show in section three below that a conceptualist interpretation of  $\varepsilon$  translates this axiom into a theorem of our extended (free) first-order logic of identity.

## 2 The Logic of Names in Conceptual Realism

The logistic framework of conceptual realism is an extension of second-order predicate logic with nominalized predicates as abstract singular terms. All predicate expressions of the system, including complex predicates generated by means of the  $\lambda$ -abstraction operator, stand for predicable concepts, which are taken to be the cognitive capacities underlying our rule-following abilities in the use of the predicates and verb phrases of natural language. Nominalized predicates, on the other hand, denote the intensional contents of the concepts that predicates stand for in their role as predicates—though some nominalized predicates, such as that for the Russell predicate  $[\lambda x(\exists F)(x = F \wedge \neg F(x))]$ , will in fact denote nothing (as a value of the bound individual variables). For this reason the first-order part of the logic, which includes the identity sign ‘=’ as a logical constant, is free of existential presuppositions.<sup>16</sup> This logic has been shown to be consistent relative to weak Zermelo set theory, and, as already noted, it is equivalent to the simple theory of types, including Leśniewski’s system of semantic categories.

---

<sup>16</sup>Of course the identity sign as used in this logic is not to be confused with the sign for strong identity in Leśniewski’s system. Because the latter is a defined, and not a primitive, constant of Leśniewski’s ontology, we will not concern ourselves with it in the remainder of this paper, i.e. hereafter we mean by ‘=’ only the identity sign of the logic of conceptual realism.

As with Leśniewski’s logistic framework, we will not concern ourselves here with the full logistic framework of conceptual realism. Rather, we need only utilize an extension of its (free) first-order logic of identity—in particular, an extension that includes quantifiers binding “nominal” variables, i.e. variables having proper and common names, vacuous or otherwise, as their substituends. Because of the connection we want to make with the nominal variables of Leśniewski’s system, we will use  $A, B, C, D$  as the name or nominal variables of conceptual realism. As individual variables, we use  $x, y, z$ , with or without numerical subscripts.

The name, or nominal, variables of conceptual realism are not individual variables, it should be noted, but represent a syntactic category distinct from that of either predicate or individual variables. The values of these variables are neither objects nor predicable concepts, but rather are concepts that are used in both general and singular reference.<sup>17</sup> The common-name concept *man*, for example, can be used in a speech act to refer to *all* men, *some* men, *most* men, *one* man, *two* men, etc., depending on the determiner or kind of referential act in which the concept is being applied. An assertion of ‘Every man is mortal’, e.g., is represented in this logic as  $(\forall xA)M(x)$ , where  $A$  is the symbolic counterpart of the common name ‘man’ and  $M$  is a one-place predicate constant for ‘is mortal’. Similarly, an assertion of ‘Some man is wise’ is represented as  $(\exists xA)W(x)$ , where  $W$  is a one-place predicate constant for ‘is wise’. The quantifier phrases  $(\forall xA)$  and  $(\exists xA)$  are taken to represent referential concepts based on the common-name concept *man*, and in particular concepts that have a structure complementary to that of predicable concepts. In conceptualism, as we understand it here, in other words, both predicable and referential concepts are unsaturated cognitive capacities that mutually saturate each other in a speech (or mental) act, the way that noun phrases and verb phrases are complementary structures that saturate each other in the formation of a sentence uttered in such an act.

An assertion of ‘Every man is mortal’ does have the same truth conditions as an assertion of ‘Everything is such that if it is a man then it is mortal’,

---

<sup>17</sup>Both proper and common names, Peter Geach has pointed out, are different from predicate expressions in that they can be “used outside the context of a sentence” in “simple acts of naming,” which are not assertions. (See [Geach 1980], p. 52.) A referential act is not an act of naming, but it does involve the use of a name together with a determiner (implicitly if not explicitly), such ‘every’, ‘some’, ‘the’, etc.

which can be symbolized as

$$(\forall x)[(\exists yA)(x = y) \rightarrow M(x)],$$

just as an assertion of ‘Some man is wise’ has the same truth conditions as ‘Something is such that it is a man and is wise’, which can be symbolized as

$$(\exists x)[(\exists yA)(x = y) \wedge W(x)].$$

Here the quantifier phrases  $(\forall x)$  and  $(\exists x)$  are read as ‘*everything*’ and ‘*something*’, i.e., they are quantifiers applied to the ultimate superordinate common name ‘thing’. For convenience, however, we adopt the usual practice of not using an explicit symbolic counterpart of the common name ‘thing’ as a part of these quantifier phrases—though it is nevertheless understood to be present. Our point about the above logical forms is that they do not represent the same speech acts as those represented by  $(\forall xA)M(x)$  and  $(\exists xA)W(x)$ . There is a difference in the cognitive structure of these assertions, and in particular in the referential and predicable concepts underlying those speech acts. One of the goals of conceptualism is to give a perspicuous logical representation of this difference in terms of a conceptualist theory of logical form. As a logistic system, in other words, conceptualism is concerned with representing both the cognitive structure of our speech (and mental) acts as well as the truth conditions of those acts in terms of the recursive operations of logical syntax. The connection between the two kinds of logical forms is given in terms of the following meaning postulates (where  $A$  is a name variable):

$$(MP1) \quad (\forall xA)\varphi \leftrightarrow (\forall x)[(\exists yA)(x = y) \rightarrow \varphi],$$

$$(MP2) \quad (\exists xA)\varphi \leftrightarrow (\exists x)[(\exists yA)(x = y) \wedge \varphi].$$

The application of  $\forall$  and  $\exists$  to *proper* names in conceptualism represents the distinction between referential acts in which a proper name is used with, as opposed to without, existential presupposition. Thus an assertion of ‘Pegasus is winged’ in which ‘Pegasus’ is used referentially without existential presupposition is symbolized as  $(\forall xB)G(x)$ , where  $B$  is a name constant for ‘Pegasus’ and  $G$  is a one-predicate for ‘is winged’. By (MP1), this assertion has the same truth conditions as ‘Anything identical with Pegasus is winged’, which in free logic does not commit us to there being anything identical to

Pegasus. An assertion of the same sentence used with existential presupposition is symbolized as  $(\exists xB)G(x)$ , which, by (MP2) has the same truth conditions as ‘Something is identical with Pegasus and is winged’.

Complex names are also part of our conceptualist framework, and are needed in the reduction of Leśniewski’s ontology. In English, complex common names are generated from more basic common names by attaching a (defining) relative clause (beginning with ‘that’, ‘who’ or ‘which’) to the latter, a syntactic device that we adopt in our logistic system as well. We use for this purpose a new primitive operator, ‘/’, which, applied to a name  $B$  and a formula  $\varphi$  results in a complex name  $B/\varphi$ , which is read as ‘ $B$  that (who, which) is  $\varphi$ ’.<sup>18</sup> Thus, an assertion of ‘Every person who works pays taxes’ in which the reference is to every person who works, and not to every person *simpliciter*, is symbolized as  $(\forall B/W(x))T(x)$ , where  $B$  is the symbolic counterpart of ‘person’,  $W$  is a one-place predicate for ‘works’, and  $T$  is a one-place predicate for ‘pays taxes’. This assertion has the same truth conditions as ‘Every person is such that if (s)he works, then (s)he pays taxes’, even though the cognitive structure of the two assertions, as based on the referential and predicable concepts of each, is different. Their equivalence is brought out in the following meaning postulate together with a similar postulate for existential quantifier phrases<sup>19</sup>:

$$(MP3) \quad (\forall xB/\varphi)\psi \leftrightarrow (\forall xB)[\varphi \rightarrow \psi],$$

$$(MP4) \quad (\exists xB/\varphi)\psi \leftrightarrow (\exists xB)[\varphi \wedge \psi].$$

We note that existential generalization, and, similarly, universal instantiation, apply to all names, whether complex or simple, proper or common, and vacuous (e.g. fictitious) or nonvacuous. Thus, if nothing is an  $A$  (e.g. a

---

<sup>18</sup>The use of ‘/’ for this purpose should not be confused with its use in representing the proper substitution in a formula of one expression for another, as in  $\varphi(x/y)$  or  $\varphi(B/C)$ . The context will make clear which use is intended. The introduction of complex names means that formulas and names are to be inductively defined simultaneously as follows: (1) every name variable (or constant) is a name; (2) where  $x, y$  are individual variables,  $(x = y)$  is a formula; and if  $\varphi, \psi$  are formulas,  $B$  is a name, and  $x$  and  $C$  are an individual and name variable, respectively, then (3)  $\neg\varphi$ , (4)  $(\varphi \rightarrow \psi)$ , (5)  $(\forall x)\varphi$ , (6)  $(\forall xB)\varphi$ , and (7)  $(\forall C)\varphi$  are formulas, and (8)  $B/\varphi$  and (9)  $/\varphi$  are (complex) names.

<sup>19</sup>Strictly speaking, both (MP2) and (MP4) are derivable in the logic from (MP1) and (MP3), respectively, because  $\exists$  is understood to be defined in terms of  $\forall$ .

unicorn), then some nominal concept  $C$  is such that nothing is a  $C$ , which can be symbolized as follows:

$$\neg(\exists y)(\exists xA)(y = x) \rightarrow (\exists C)\neg(\exists y)(\exists xC)(y = x).$$

It does not follow, of course, that such an inference requires us to give only a substitutional interpretation to quantifiers binding nominal variables. The following will be taken as the axioms for nominal quantifiers<sup>20</sup>:

**Axiom 1:**  $(\forall C)\varphi \rightarrow \varphi(B[x]/C)$ , *where  $B$  is free for  $C$  in  $\varphi$  with respect to  $x$ ,*

**Axiom 2:**  $(\forall C)[\varphi \rightarrow \psi] \rightarrow [(\forall C)\varphi \rightarrow (\forall C)\psi]$ ,

**Axiom 3:**  $[\varphi \rightarrow (\forall C)\varphi]$ , *where  $C$  is not free in  $\varphi$ .*

In addition to these axioms, we assume all tautologous formulas and the quantifier and identity axioms for first-order logic free of existential presuppositions.<sup>21</sup> As inference rules we assume modus ponens and the rule of

---

<sup>20</sup>We assume the obvious definitions of bondage and freedom of occurrences of nominal variables in formulas, and of the proper substitution in a formula  $\varphi$  of a nominal variable or constant  $B$  for free occurrences of a nominal variable  $C$ . A complex name  $B/\xi$  is free for  $C$  in  $\varphi$  with respect to an individual variable  $x$  (as place holder) if for each variable  $y$  such that  $(\forall yC)$  occurs in  $\varphi$  and  $C$  is free at that occurrence, then  $y$  is free for  $x$  in  $B/\xi$ , and, moreover, that no variable, nominal or individual, other than  $x$  that is free in  $B/\xi$  becomes bound when that free occurrence of  $C$  in  $\varphi$  is replaced by an occurrence of  $B/\xi(y/x)$ . In Axiom 1, it is understood that if  $B$  is a complex name, then it is free for  $C$  in  $\varphi$  with respect to a particular variable  $x$  and that  $\varphi(B[x]/C)$  represents the result of replacing free occurrences of  $C$  by free occurrences of  $B$  with respect to  $x$ .

<sup>21</sup>With only  $\neg$ ,  $\rightarrow$ ,  $=$ , and  $\forall$  as primitive logical constants (i.e.  $\wedge$ ,  $\vee$ ,  $\leftrightarrow$ , and  $\exists$  are assumed defined in the usual way), we can take the following as the quantifier and identity axiom schemas for our free logic:

$$\begin{aligned} (\forall x)[\varphi \rightarrow \psi] &\rightarrow [(\forall x)\varphi \rightarrow (\forall x)\psi], \\ \chi &\rightarrow (\forall x)\chi, \\ (\forall x)(\exists y)(x = y), \\ x &= x, \\ x = y &\rightarrow [\varphi \rightarrow \psi], \end{aligned}$$

where  $x$  is not free in  $\chi$ , and where  $\psi$  is obtained from  $\varphi$  by replacing one or more free occurrences of  $x$  by a free occurrence of  $y$ .

universal generalization, (UG), for individual and nominal variables. The rule (UG) for restricted quantifiers, i.e.,

$$\text{if } \vdash \varphi, \text{ then } \vdash (\forall xA)\varphi,$$

is derivable from (MP1) and the rule (UG). From these axioms and rules it is easy to prove the rules for interchanging provably equivalent formulas and for rewriting bound individual variables. Where  $y$  is free for  $x$  in  $\varphi$ , the following qualified version of universal instantiation is provable on the basis of Leibniz's law (an axiom), universal generalization, and the quantifier axioms:

$$\text{T1: } \vdash (\forall x)\varphi \rightarrow [(\exists z)(y = z) \rightarrow \varphi(y/x)].$$

Some useful theorems regarding the nominal quantifiers are:

$$\text{T2: } \vdash (\forall x)\varphi \rightarrow (\forall xA)\varphi,$$

which is derivable from (MP1) and tautologous transformations. By taking  $\varphi$  in T2 to be  $(y \neq z)$ , we have:

$$\text{T3: } \vdash (\exists zA)(y = z) \rightarrow (\exists z)(y = z).$$

From this theorem, the rewrite of bound variables rule, and (MP1),

$$\text{T4: } \vdash (\forall xA)\varphi \rightarrow [(\exists zA)(y = z) \rightarrow \varphi(y/x)],$$

then follows, where, again,  $y$  is free for  $x$  in  $\varphi$ . Finally, the following theorem indicates how unrestricted quantification (over *things*) is related to nominal quantification:

$$\text{T5: } \vdash (\forall x)\varphi \leftrightarrow (\forall A)(\forall xA)\varphi,$$

where the nominal (name) variable  $A$  is not free in  $\varphi$ . The left-to-right direction follows by T2, (UG), and the quantifier axioms for names. The right-to-left direction follows by first universally instantiation  $A$  to *thing identical to itself*, i.e. to  $/x = x$ , so that by Axiom 1 we have:

$$\vdash (\forall A)(\forall xA)\varphi \rightarrow (\forall x/x = x)\varphi.$$

But, by (MP3),

$$\vdash (\forall x/x = x)\varphi \rightarrow (\forall x)[x = x \rightarrow \varphi],$$

and from this and the axioms for identity, (UG), and the absolute quantifier,

$$\vdash (\forall x/x = x)\varphi \rightarrow (\forall x)\varphi,$$

from which the right-to-left direction of T5 follows.

### 3 A Conceptualist Interpretation of Leśniewski's System

We now turn to a translation of the first-order version of Leśniewski's ontology as formulated in section 1. We assume that the nominal variables  $a, b, c, d$  (with or without numerical subscripts) of this system are correlated one-to-one with the nominal or name variables  $A, B, C, D$  (with or without numerical subscripts) of our extended first-order logic of identity that is free of existential presuppositions; i.e., that  $A$  is correlated with  $a$ ,  $B$  is correlated with  $b$ , etc. Because the only atomic formulas of the system are of the form ' $a \varepsilon b$ ', the following inductive definition of the translation function  $trs$  translates each formula of Leśniewski's system into a formula of our alternative system (with  $a$  replaced by  $A$ ,  $b$  by  $B$ , etc.):

1.  $trs(a \varepsilon b) = (\forall xA)(\forall yA)(x = y) \wedge (\exists xA)(\exists yB)(x = y)$ ,
2.  $trs(\neg\varphi) = \neg trs(\varphi)$ ,
3.  $trs(\varphi \rightarrow \psi) = [trs(\varphi) \rightarrow trs(\psi)]$ ,
4.  $trs((\forall a)\varphi) = (\forall A)trs(\varphi)$ .

Note that an atomic formula  $a \varepsilon b$  is interpreted here as stating that at most one thing is  $a$  and that some  $a$  is a  $b$ , which has the same truth conditions as saying that exactly one thing is  $a$  and that thing is a  $b$ . It is clear that where  $\varphi$  is a logical axiom of the first-order logic of Leśniewski's ontology, then  $trs(\varphi)$  is a theorem of our extended free first-order logic of identity.<sup>22</sup> Modus ponens and (UG) in all its forms also preserve validity under  $trs$ . Accordingly, to show that this interpretation amounts to a reduction of Leśniewski's ontology, we need only prove that  $trs$  translates Leśniewski's single axiom for his ontology into a theorem of our present system. We will deal first with the translation of the left-to-right direction of this axiom, the consequent of which consists of three conjuncts. For this direction, we need only show that the translation of  $a \varepsilon b$  yields the translation of each conjunct, which is established in T8, T10 and T12 below.

---

<sup>22</sup>In fact, Axioms 1-3 are just the translations of the quantifier axioms assumed in the first-order theory for Leśniewski's ontology (as described in footnote 8). By definition of  $trs$ , moreover, it is obvious that the translation of a tautologous formula is also a tautologous formula.

**T6:**  $\vdash (\exists xA)\varphi \rightarrow (\exists xA)(\exists yA)(x = y)$ .

**Proof.** By sentential logic,  $\vdash (x \neq y) \rightarrow [(x = y) \rightarrow \neg\varphi]$ , and therefore, by (UG) and the quantifier axioms, where  $y$  is not free in  $\varphi$ ,  $\vdash (\forall yA)(x \neq y) \rightarrow [(\exists yA)(x = y) \rightarrow \neg\varphi]$ . Therefore, by sentential logic,  $\vdash [(\exists yA)(x = y) \rightarrow (\forall yA)(x \neq y)] \rightarrow [(\exists yA)(x = y) \rightarrow [(\exists yA)(x = y) \rightarrow \neg\varphi]]$ , from which, by tautologous transformations, (UG) on  $x$ , quantifier logic and (MP1) twice, it follows that  $\vdash (\forall xA)(\forall yA)(x \neq y) \rightarrow (\forall xA)\neg\varphi$ , and hence  $\vdash (\exists xA)\varphi \rightarrow (\exists xA)(\exists yA)(x = y)$ . ■

**T7:**  $\vdash (\exists xA)(\exists yB)(x = y) \rightarrow (\exists xA)(\exists yA)(x = y)$ .

**Proof.** By T6, where  $\varphi$  is  $(\exists yB)(x = y)$ . ■

**T8:**  $\vdash \text{trs}(a \varepsilon b \rightarrow (\exists c)c \varepsilon a)$ , i.e.,

$\vdash (\forall xA)(\forall yA)(x = y) \wedge (\exists xA)(\exists yB)(x = y) \rightarrow (\exists C)[(\forall xC)(\forall yC)(x = y) \wedge (\exists xC)(\exists yA)(x = y)]$ .

**Proof.** By T7, sentential logic and existential generalization of the nominal variable  $A$ . ■

**T9:**  $\vdash (\forall xA)(\forall yA)(x = y) \wedge (\exists xA)(\exists yB)(x = y) \rightarrow [(\exists yA)(x = y) \rightarrow (\exists yB)(x = y)]$ .

**Proof.** By T4, Leibniz's law, and sentential logic,  $\vdash (\forall yA)(x = y) \wedge (\exists yA)(z = y) \wedge (\exists yB)(z = y) \rightarrow (x = z) \wedge (\exists yB)(x = y)$ , and therefore, by (UG) on  $z$  and quantifier logic,  $\vdash (\forall yA)(x = y) \wedge (\exists z)[(\exists yA)(z = y) \wedge (\exists yB)(z = y)] \rightarrow (\exists yB)(x = y)$ , and hence, by (MP2),  $\vdash (\forall yA)(x = y) \wedge (\exists zA)(\exists yB)(z = y) \rightarrow (\exists yB)(x = y)$ , which, by rewrite of bound variables, shows that  $\vdash (\forall yA)(x = y) \wedge (\exists xA)(\exists yB)(x = y) \rightarrow (\exists yB)(x = y)$ . But, by T4,  $\vdash (\forall xA)(\forall yA)(x = y) \rightarrow [(\exists yA)(x = y) \rightarrow (\forall yA)(x = y)]$ , and therefore by tautologous transformations,  $\vdash (\forall xA)(\forall yA)(x = y) \wedge (\exists xA)(\exists yB)(x = y) \rightarrow [(\exists yA)(x = y) \rightarrow (\exists yB)(x = y)]$ . ■

**T10:**  $\vdash \text{trs}(a \varepsilon b \rightarrow (\forall c)[c \varepsilon a \rightarrow c \varepsilon b])$ , i.e.,

$\vdash (\forall xA)(\forall yA)(x = y) \wedge (\exists xA)(\exists yB)(x = y) \rightarrow (\forall C)[(\forall xC)(\forall yC)(x = y) \wedge (\exists xC)(\exists yA)(x = y) \rightarrow (\forall xC)(\forall yC)(x = y) \wedge (\exists xC)(\exists yB)(x = y)]$ .

**Proof.** By T9, (UG) on  $x$  and quantifier laws for  $(\forall xC)$ , and then by sentential logic, (UG) and quantifier laws for  $C$ . ■

**T11:**  $\vdash (\forall xA)(\forall yA)(x = y) \rightarrow [(\exists xC)(\exists yA)(x = y) \wedge (\exists xD)(\exists yA)(x = y) \rightarrow (\exists xC)(\exists yD)(x = y)]$ .

**Proof.** By T9, substituting  $D$  for  $B$ ,  $\vdash (\forall xA)(\forall yA)(x = y) \wedge (\exists xA)(\exists yD)(x = y) \rightarrow [(\exists yA)(x = y) \rightarrow (\exists yD)(x = y)]$ . But, by commutation and identity laws,  $\vdash (\exists yA)(\exists xD)(y = x) \rightarrow (\exists xD)(\exists yA)(x = y)$ , and hence  $\vdash (\forall xA)(\forall yA)(x = y) \rightarrow [(\exists yA)(x = y) \wedge (\exists xD)(\exists yA)(x = y) \rightarrow (\exists yD)(x = y)]$ . T11 then follows by (UG) on  $x$  and quantifier laws for  $(\forall xC)$ .

■  
**T12:**  $\vdash trs(a \varepsilon b \rightarrow (\forall c)(\forall d)[c \varepsilon a \wedge d \varepsilon a \rightarrow c \varepsilon d])$ , i.e.,  $\vdash (\forall xA)(\forall yA)(x = y) \wedge (\exists xA)(\exists yB)(x = y) \rightarrow (\forall C)(\forall D)[(\forall xC)(\forall yC)(x = y) \wedge (\exists xC)(\exists yA)(x = y) \wedge (\forall xD)(\forall yD)(x = y) \wedge (\exists xD)(\exists yA)(x = y) \rightarrow (\forall xC)(\forall yC)(x = y) \wedge (\exists xC)(\exists yD)(x = y)]$ .

**Proof.** By T11, tautologous transformations, (UG) with respect to  $C$  and  $D$ , and quantifier laws for  $(\forall C)$  and  $(\forall D)$ . ■

By T8, T10, and T12, the translation of the left-to-right direction of Leśniewski's axiom for ontology is provable in our extended identity logic. We will break down our proof of the converse direction into two parts, each dealing with one of the two conjuncts in our translation of ' $a \varepsilon b$ '. In the first part, which is T15 below, we show that

$$\vdash trs((\exists c)c \varepsilon a \wedge (\forall c)[c \varepsilon a \rightarrow c \varepsilon b]) \rightarrow (\exists xA)(\exists yB)(x = y),$$

and in the second, which is T22 below, we show that

$$\vdash trs((\exists c)c \varepsilon a \wedge (\forall c)(\forall d)[c \varepsilon a \wedge d \varepsilon a \rightarrow c \varepsilon d]) \rightarrow (\forall xA)(\forall yA)(x = y).$$

**T13:**  $\vdash (\forall yC)(x = y) \rightarrow (\forall xC)(\forall yC)(x = y)$ .

**Proof.** By T4, (UG) on  $y$  and quantifier laws for  $(\forall yC)$ ,  $\vdash (\forall yC)(x = y) \rightarrow [(\exists yC)(z = y) \rightarrow (\forall yC)(z = y)]$ , and therefore, by (UG) on  $z$  and quantifier laws for  $(\forall z)$ ,  $\vdash (\forall yC)(x = y) \rightarrow (\forall zC)(\forall yC)(z = y)$ , from which T13 follows by the rewrite law for bound variables. ■

**T14:**  $\vdash (\exists x)[(\forall yC)(x = y) \wedge (\exists yC)(x = y) \wedge (\exists yA)(x = y)] \wedge (\forall C)[(\forall xC)(\forall yC)(x = y) \wedge (\exists xC)(\exists yA)(x = y) \rightarrow (\exists xC)(\exists yB)(x = y)] \rightarrow (\exists xA)(\exists yB)(x = y)$ .

**Proof.** By T4,  $\vdash (\forall yC)(x = y) \rightarrow [(\exists yC)(z = y) \rightarrow x = z]$ , and by Leibniz's law,  $\vdash (\exists yB)(z = y) \rightarrow [x = z \rightarrow (\exists yB)(x = y)]$ , and hence, by (UG) on  $z$ , tautologous transformations and quantifier laws for  $z$ ,  $\vdash (\forall yC)(x = y) \rightarrow [(\exists z)[(\exists yC)(z = y) \wedge (\exists yB)(z = y)] \rightarrow (\exists yB)(x = y)]$ .

But then, by (MP2) and the rewrite law for bound variables,  $\vdash (\forall yC)(x = y) \rightarrow [(\exists xC)(\exists yB)(x = y) \rightarrow (\exists yB)(x = y)]$ , from which it follows by sentential logic that  $\vdash (\forall yC)(x = y) \wedge (\exists yC)(x = y) \wedge (\exists yA)(x = y) \rightarrow [(\exists xC)(\exists yB)(x = y) \rightarrow (\exists yA)(x = y) \wedge (\exists yB)(x = y)]$ . Accordingly, by (UG) and quantifier laws for  $x$ ,  $\vdash (\exists x)[(\forall yC)(x = y) \wedge (\exists yC)(x = y) \wedge (\exists yA)(x = y)] \rightarrow [(\exists xC)(\exists yB)(x = y) \rightarrow (\exists x)[(\exists yA)(x = y) \wedge (\exists yB)(x = y)]]$ . Finally, we note that by (MP2),  $\vdash (\exists xC)(\exists yA)(x = y) \leftrightarrow (\exists x)[(\exists yC)(x = y) \wedge (\exists yA)(x = y)]$ , and therefore, by T13 and universal instantiation of  $C$ ,  $\vdash (\exists x)[(\forall yC)(x = y) \wedge (\exists yC)(x = y) \wedge (\exists yA)(x = y)] \wedge (\forall C)[(\forall xC)(\forall yC)(x = y) \wedge (\exists xC)(\exists yA)(x = y)] \rightarrow (\exists xC)(\exists yB)(x = y)$ , from which T14 follows. ■

**T15:**  $\vdash trs((\exists c)c \varepsilon a \wedge (\forall c)[c \varepsilon a \rightarrow c \varepsilon b]) \rightarrow (\exists xA)(\exists yB)(x = y)$ , i.e.,  
 $\vdash (\exists C)[(\forall xC)(\forall yC)(x = y) \wedge (\exists xC)(\exists yA)(x = y)] \wedge (\forall C)[(\forall xC)(\forall yC)(x = y) \wedge (\exists xC)(\exists yA)(x = y) \rightarrow (\forall xC)(\forall yC)(x = y) \wedge (\exists xC)(\exists yB)(x = y)] \rightarrow (\exists xA)(\exists yB)(x = y)$ .

**Proof.** By (MP2), T4, and quantifier logic,  $\vdash (\forall xC)(\forall yC)(x = y) \wedge (\exists xC)(\exists yA)(x = y) \rightarrow (\exists x)[(\exists yC)(x = y) \wedge (\forall yC)(x = y) \wedge (\exists yA)(x = y)]$ , from which, together T14, T15 follows. ■

We now turn to showing the translation of the second part of the right-to-left direction of Leśniewski's axiom. Some useful theorems for this purpose are the following.

**T16:**  $\vdash (\forall x)\varphi \leftrightarrow (\forall x)[(\exists y)(x = y) \rightarrow \varphi]$ .

**Proof.** By T1, (UG) on  $x$ , and identity and quantifier axioms. ■

**T17:**  $\vdash (\exists x)\varphi \leftrightarrow (\exists x)[(\exists y)(x = y) \wedge \varphi]$ .

**Proof.** By T16, definition of  $\exists$ , and tautologous transformations. ■

**T18:**  $\vdash (\forall x/x = z)\varphi \leftrightarrow (\forall x)[x = z \rightarrow \varphi]$ .

**Proof.** By (MP3),  $\vdash (\forall x/x = z)\varphi \leftrightarrow (\forall x)[(\exists y/y = z)(x = y) \rightarrow \varphi]$ , and by (MP4),  $\vdash (\exists y/y = z)(x = y) \leftrightarrow (\exists y)[(\exists w)(y = w) \wedge (y = z) \wedge (x = z)]$ , and therefore by T17 and Leibniz's law,  $\vdash (\exists y/y = z)(x = y) \leftrightarrow (\exists y)(x = y) \wedge (x = z)$ . Hence,  $\vdash (\forall x)[(\exists y/y = z)(x = y) \rightarrow \varphi] \leftrightarrow (\forall x)[(\exists y)(x = y) \rightarrow (x = z \rightarrow \varphi)]$ , and therefore, by quantifier axioms,  $\vdash \forall x[(\exists y/y = z)(x = y) \rightarrow \varphi] \rightarrow [(\forall x)(\exists y)(x = y) \rightarrow (\forall x)(x = z \rightarrow \varphi)]$ , which, by an identity axiom, yields the left-to-right direction of T18. For the converse direction, note that by sentential logic,  $\vdash (\forall x)[x = z \rightarrow \varphi] \rightarrow (\forall x)[(\exists y)(x = y) \rightarrow (x = z \rightarrow \varphi)]$ , and therefore, by above biconditional,  $\vdash (\forall x)[x = z \rightarrow \varphi] \rightarrow (\forall x)[(\exists y/y = z)$

$z)(x = y) \rightarrow \varphi]$ , and hence  $\vdash (\forall x)[x = z \rightarrow \varphi] \rightarrow (\forall x/x = z)\varphi$ , which is the right-to-left direction of T18. ■

**T19:**  $\vdash (\forall x/x = z)(\forall y/y = z)(x = y)$ .

**Proof.** By T18 twice,  $\vdash (\forall x/x = z)(\forall y/y = z)(x = y) \leftrightarrow (\forall x)[x = z \rightarrow (\forall y)(y = z \rightarrow x = y)]$ , from which T19 follows by identity logic. ■

**T20:**  $\vdash (\exists x/x = z)(\exists yA)(x = y) \leftrightarrow (\exists yA)(z = y)$ .

**Proof.** By (MP4),  $\vdash (\exists x/x = z)(\exists yA)(x = y) \leftrightarrow (\exists x)[(x = z) \wedge (\exists yA)(x = y)]$ , and therefore, by quantifier axioms and Leibniz's law,  $\vdash (\exists x/x = z)(\exists yA)(x = y) \rightarrow (\exists yA)(z = y)$ , which is the left-to-right direction of T20. For the converse direction, note that, by (MP2),  $\vdash (\exists yA)(z = y) \leftrightarrow (\exists y)[(\exists xA)(y = x) \wedge (z = y)]$ , and hence, by rewrite of bound variables,  $\vdash (\exists yA)(z = y) \leftrightarrow (\exists x)[(\exists yA)(x = y) \wedge (z = x)]$ , from which, by identity logic, it follows that  $\vdash (\exists yA)(z = y) \rightarrow (\exists x)[(x = z) \wedge (\exists yA)(x = y)]$ ; and therefore, by (MP4),  $\vdash (\exists yA)(z = y) \rightarrow (\exists x/x = z)(\exists yA)(x = y)$ . ■

**T21:**  $\vdash (\forall D)[(\forall xD)(\forall yD)(x = y) \wedge (\exists xD)(\exists yA)(x = y) \rightarrow (\forall xC)(\forall yC)(x = y) \wedge (\exists xD)(\exists yC)(x = y)] \rightarrow (\forall xA)(\forall yA)(x = y)$ .

**Proof.** By Axiom 1, substituting ‘thing identical to  $x_0$ ’, i.e.  $/z = x_0$  for  $D$  with respect to the variable  $z$ ,  $\vdash (\forall D)[(\forall xD)(\forall yD)(x = y) \wedge (\exists xD)(\exists yA)(x = y) \rightarrow (\forall xC)(\forall yC)(x = y) \wedge (\exists xD)(\exists yC)(x = y)] \rightarrow [(\forall x/x = x_0)(\forall y/y = x_0)(x = y) \wedge (\exists x/x = x_0)(\exists yA)(x = y) \rightarrow (\forall xC)(\forall yC)(x = y) \wedge (\exists x/x = x_0)(\exists yC)(x = y)]$ . But, by T19,  $\vdash (\forall x/x = x_0)(\forall y/y = x_0)(x = y)$ , and, by T20 (twice),  $\vdash (\exists x/x = x_0)(\exists yA)(x = y) \leftrightarrow (\exists yA)(x_0 = y)$  and  $\vdash (\exists x/x = x_0)(\exists yC)(x = y) \leftrightarrow (\exists yC)(x_0 = y)$ , and therefore  $\vdash (\forall D)[(\forall xD)(\forall yD)(x = y) \wedge (\exists xD)(\exists yA)(x = y) \rightarrow (\forall xC)(\forall yC)(x = y) \wedge (\exists xD)(\exists yC)(x = y)] \rightarrow [(\exists yA)(x_0 = y) \rightarrow (\forall xC)(\forall yC)(x = y) \wedge (\exists yC)(x_0 = y)]$ . Also, by substituting  $y_0$  for  $x_0$  in this last theorem,  $\vdash (\forall D)[(\forall xD)(\forall yD)(x = y) \wedge (\exists xD)(\exists yA)(x = y) \rightarrow (\forall xC)(\forall yC)(x = y) \wedge (\exists xD)(\exists yC)(x = y)] \rightarrow [(\exists yA)(y_0 = y) \rightarrow (\forall xC)(\forall yC)(x = y) \wedge (\exists yC)(y_0 = y)]$ , and hence  $\vdash (\forall D)[(\forall xD)(\forall yD)(x = y) \wedge (\exists xD)(\exists yA)(x = y) \rightarrow (\forall xC)(\forall yC)(x = y) \wedge (\exists xD)(\exists yC)(x = y)] \rightarrow [(\exists yA)(x_0 = y) \wedge (\exists yA)(y_0 = y) \rightarrow x_0 = y_0]$ . Then, by (UG) on  $x_0$  and  $y_0$ , and quantifier axioms for  $x_0$  and  $y_0$ ,  $\vdash (\forall D)[(\forall xD)(\forall yD)(x = y) \wedge (\exists xD)(\exists yA)(x = y) \rightarrow (\forall xC)(\forall yC)(x = y) \wedge (\exists xD)(\exists yC)(x = y)] \rightarrow (\forall x_0)[(\exists yA)(x_0 = y) \rightarrow (\forall y_0)[(\exists yA)(y_0 = y) \rightarrow x_0 = y_0]]$ , from which T21 follows by (MP1) and the rewrite law for bound variables. ■

**T22:**  $\vdash trs((\exists c)c \varepsilon a \wedge (\forall c)(\forall d)[c \varepsilon a \wedge d \varepsilon a \rightarrow c \varepsilon d]) \rightarrow (\forall xA)(\forall yA)(x = y)$ ,  
i.e.,  $\vdash (\exists C)[(\forall xC)(\forall yC)(x = y) \wedge (\exists xC)(\exists yA)(x = y)] \wedge$   
 $(\forall C)(\forall D)[(\forall xC)(\forall yC)(x = y) \wedge (\exists xC)(\exists yA)(x = y) \wedge (\forall xD)(\forall yD)(x =$   
 $y) \wedge (\exists xD)(\exists yA)(x = y) \rightarrow (\forall xC)(\forall yC)(x = y) \wedge (\exists xC)(\exists yD)(x = y)] \rightarrow$   
 $(\forall xA)(\forall yA)(x = y)$ .

**Proof.** By Axiom 1,  $\vdash (\forall xC)(\forall yC)(x = y) \wedge (\exists xC)(\exists yA)(x = y)] \wedge$   
 $(\forall C)(\forall D)[(\forall xC)(\forall yC)(x = y) \wedge (\exists xC)(\exists yA)(x = y) \wedge (\forall xD)(\forall yD)(x =$   
 $y) \wedge (\exists xD)(\exists yA)(x = y) \rightarrow (\forall xC)(\forall yC)(x = y) \wedge (\exists xC)(\exists yD)(x = y)] \rightarrow$   
 $(\forall xC)(\forall yC)(x = y) \wedge (\forall D)[(\forall xD)(\forall yD)(x = y) \wedge (\exists xD)(\exists yA)(x = y)] \rightarrow$   
 $(\forall xC)(\forall yC)(x = y) \wedge (\exists xC)(\exists yD)(x = y)]$ . But, by commutation law,  
 $\vdash (\exists xC)(\exists yD)(x = y) \leftrightarrow (\exists xD)(\exists yC)(x = y)$ , and therefore, by T21,  
 $\vdash (\forall xC)(\forall yC)(x = y) \wedge (\exists xC)(\exists yA)(x = y)] \wedge (\forall C)(\forall D)[(\forall xC)(\forall yC)(x =$   
 $y) \wedge (\exists xC)(\exists yA)(x = y) \wedge (\forall xD)(\forall yD)(x = y) \wedge (\exists xD)(\exists yA)(x = y) \rightarrow$   
 $(\forall xC)(\forall yC)(x = y) \wedge (\exists xC)(\exists yD)(x = y)] \rightarrow (\forall xA)(\forall yA)(x = y)$ , from  
which, by (UG) and quantifier laws on  $C$ , T22 follows. ■

By T15 and T22, we now have as theorems both the left-to-right and the right-to-left directions of our translation of Leśniewski's axiom for ontology. We conclude, accordingly, that Leśniewski's axiom is a theorem of our system under the translation  $trs$ , and hence that the translation of every theorem of Leśniewski's first-order theory of ontology is a theorem of our system.

**Metatheorem 1:** If  $\varphi$  is a theorem of Leśniewski's first-order theory of ontology based on the relation of singular inclusion (but not also his theory of creative definitions of name-forming functors), then  $trs(\varphi)$  is a theorem of the free first-order logic of identity extended to include nominal quantifiers.

## 4 Reduction of Leśniewski's Theory of Definitions

The parenthetical qualification in metatheorem 1 above can be dropped, it turns out, because there is an extension of the first-order version of Leśniewski's ontology described in section one in which Leśniewski's theory of definitions can be shown to be noncreative, and that extension is also reducible to our extended first-order logic of identity. The extension in question has been referred to as Leśniewski's "elementary ontology," or more simply as the sys-

tem  $EO$ .<sup>23</sup> This system is obtained from the first-order theory described in section one by adding the following two axioms (which are special cases of a more general comprehension principle, which is derivable in  $EO$ ):

$$\text{(Compl)} \quad (\forall a)(\exists b)(\forall c)[c\epsilon b \leftrightarrow c\epsilon c \wedge c\epsilon a],$$

$$\text{(Conj)} \quad (\forall a)(\forall b)(\exists c)(\forall d)[d\epsilon c \leftrightarrow d\epsilon a \wedge d\epsilon b].$$

These axioms, expressed in terms of our conceptualist logic, posit the existence of a complementary nominal concept corresponding to any given nominal concept  $A$  (which can be read in English as ‘thing that is not an  $A$ ’) as well as the existence of a conjunctive nominal concept determined by arbitrary nominal concepts  $A$  and  $B$  (which can be read in English as ‘thing that is both an  $A$  and a  $B$ ’). The Translations of both (Compl) and (Conj) are derivable from special instances of the following comprehension principle:

$$\text{(CP)} \quad (\forall A)(\exists B)(\forall x)[(\exists yB)(x = y) \leftrightarrow (\exists yA)(x = y)].$$

This comprehension principle is immediately derivable from Axiom 1 (or rather from the contrapositive of Axiom 1, which amounts to a form of existential generalization for nominal concepts). By (CP) and Axiom 1, substituting for  $A$  the complex name form  $\neg(\exists zA)(y = z)$  (which is read as ‘thing that is not an  $A$ ’), we have

$$\vdash (\exists B)(\forall x)[(\exists yB)(x = y) \leftrightarrow (\exists y/\neg(\exists zA)(y = z))(x = y)],$$

from which, by (MP4) and Leibniz’s law, it follows that

$$\vdash (\exists B)(\forall x)[(\exists yB)(x = y) \leftrightarrow \neg(\exists zA)(x = z)],$$

which affirms the existence of a nominal concept that is the complement of  $A$ . Finally, by proofs similar to those given in section three, it can be shown on the basis of this last theorem that the translation of (Compl), i.e.  $trs(\text{Compl})$ , is provable in our conceptualist logic of identity.

---

<sup>23</sup>See [Iwanuś 1973]. Strictly speaking,  $EO$  also contains several noncreative definitional axioms that are justified (eliminable) on the basis of the two axioms listed here. The translations of each of these definitions is also provable in our extended first-order conceptualist logic of identity.

Similarly, by (rewriting  $B$  to  $C$  in) (CP) and Axiom 1, substituting now for  $A$  the complex name form  $\lambda z(\exists zA)(y = z) \wedge (\exists zB)(y = z)$  (which is read as ‘thing that is both an  $A$  and a  $B$ ’), we have

$$\vdash (\exists C)(\forall x)[(\exists yC)(x = y) \leftrightarrow (\exists y/(\exists zA)(y = z) \wedge (\exists zB)(y = z))(x = y)],$$

which, by (MP4), reduces to

$$\vdash (\exists C)(\forall x)[(\exists yC)(x = y) \leftrightarrow (\exists y)[(\exists zA)(y = z) \wedge (\exists zB)(y = z) \wedge y = x]],$$

and hence, by Leibniz’s law, to

$$\vdash (\exists C)(\forall x)[(\exists yC)(x = y) \leftrightarrow (\exists zA)(x = z) \wedge (\exists zB)(x = z)],$$

which affirms the existence of a nominal concept corresponding to the conjunction of being both an  $A$  and a  $B$ . Again, by proofs similar to those in section three, it can be shown on the basis of this last theorem that the translation of (Conj), i.e.  $trs(\text{Conj})$ , is provable in our conceptualist logic of identity.

**Metatheorem 2:** If  $\varphi$  is a theorem of  $EO$ , Leśniewski’s system of elementary ontology (including his theory of definitions), then  $trs(\varphi)$  is a theorem of the free first-order logic of identity extended to include nominal quantifiers.

## 5 Consistency and Decidability

It has been shown in [Iwanus 1973] that Leśniewski’s system of elementary ontology is decidable, i.e., that it is decidable for any formula of  $EO$  whether or not it is a theorem of  $EO$ . We will show here that our present conceptualist system to which  $EO$  has been reduced is also decidable as well as consistent. First, let us note that by (MP3) and a simple inductive argument it can be shown that every formula of our conceptualist system in which a complex name occurs is provably equivalent to a formula in which no complex name occurs.

**Metatheorem 3:** If  $\varphi$  is a formula of our free first-order logic extended to include nominal (name) variables, quantification over such, and restricted quantifiers with respect to such, then there is a formula  $\psi$  in which no complex

name occurs such that  $\varphi$  is provably equivalent to  $\psi$  in this logic, i.e.,  $\vdash \varphi \leftrightarrow \psi$ .

Because of the above metatheorem, we can, in what follows, restrict ourselves to formulas in which no complex name occurs.<sup>24</sup> We assume a one-to-one correlation of the nominal variables  $A, B, C, D$ , etc., with one-place predicate variables  $F_A, F_B, F_C, F_D$ , etc., and inductively define a translation function  $trs^*$  from the formulas of our extended free first-order logic of identity in which no complex name occurs into formulas of second-order monadic predicate logic (with identity) as follows:

1.  $trs^*(x = y) = (x = y)$ ,
2.  $trs^*(\neg\varphi) = \neg trs^*(\varphi)$ ,
3.  $trs^*(\varphi \rightarrow \psi) = [trs^*(\varphi) \rightarrow trs^*(\psi)]$ ,
4.  $trs^*(\forall x\varphi) = (\forall x)trs^*(\varphi)$ ,
5.  $trs^*(\forall xA)\varphi = (\forall x)[F_A(x) \rightarrow \varphi]$ ,
6.  $trs^*(\forall A)\varphi = (\forall F_A)trs^*(\varphi)$ .

It is clear that the translation under  $trs^*$  of every theorem of our extended free first-order logic of identity with nominal quantifiers in which no complex name occurs becomes a theorem of second-order monadic predicate logic. The restriction to formulas in which no complex names occur can be dropped by allowing, for each formula  $\varphi$  in which a complex name does occur, the translation function  $trs^*$  to assign  $trs^*(\psi)$  to  $\varphi$ , where  $\psi$  is the first formula (in terms of some alphabetic ordering) in which no complex name occurs and such that  $\vdash \varphi \leftrightarrow \psi$ . By extending  $trs^*$  in this way, it then follows that every theorem of our conceptualist logic, i.e., our extended free first-order logic of identity, is translated into a theorem of second-order monadic predicate logic, and hence, given the known consistency of the latter, that our conceptualist system is consistent. Accordingly, by metatheorem 2, it follows that the product translation function,  $trs/trs^*$ , translates every theorem of Leśniewski's elementary ontology into a theorem of second-order monadic predicate logic, which, as is well-known, is decidable.<sup>25</sup>

---

<sup>24</sup>We also ignore name constants and concern ourselves only with formulas in which no applied descriptive constants occur.

<sup>25</sup>See, e.g., [Church 1956], p. 303.

**Metatheorem 4:** If  $\varphi$  is a theorem of our present conceptualist logic, then  $trs^*(\varphi)$  is a theorem of second-order monadic predicate logic; and, similarly, if  $\varphi$  is a theorem of *EO*, then  $trs^*(trs(\varphi))$  is also a theorem of second-order monadic predicate logic. Therefore, both our present conceptualist logic and *EO* are consistent.

To prove the decidability of our conceptualist logic we first note that every theorem of second-order monadic predicate logic can be translated into a theorem of our conceptualist logic, and hence that the one system is essentially equivalent to the other. Because monadic predicate variables are in a one-to-one correspondence with nominal variables, we can take each predicate variable to have the form  $F_A$ , where  $A$  is the nominal variable corresponding to that predicate variable. The translation function,  $trs'$ , is defined as follows:

1.  $trs'(x = y) = (x = y)$ ,
2.  $trs'(F_A(x)) = (\exists yA)(x = y)$ , where  $y$  is the first individual variable other than  $x$ ,
3.  $trs'(\neg\varphi) = \neg trs'(\varphi)$ ,
4.  $trs'(\varphi \rightarrow \psi) = (trs'(\varphi) \rightarrow trs'(\psi))$ ,
5.  $trs'((\forall x)\varphi) = (\forall x)trs'(\varphi)$ ,
6.  $trs'((\forall F_A)\varphi) = (\forall A)trs'(\varphi)$ .

It is easily seen that the translation under  $trs'$  of each axiom of second-order monadic predicate logic is a theorem of our conceptualist system. The comprehension principle, for example, which is formulated as follows:

$$(\exists F_A)(\forall x)[F_A(x) \leftrightarrow \varphi],$$

where  $F_A$  is not free in  $\varphi$ , and which can be taken as an axiom instead of the universal instantiation law, has the following as its translation:

$$(\exists A)(\forall x)[(\exists yA)(x = y) \leftrightarrow trs'(\varphi)].$$

This formula can be derived from the following instance of (CP), the comprehension principle of our conceptualist logic:

$$(\exists A)(\forall x)[(\exists yA)(x = y) \leftrightarrow (\exists y/trs'(\varphi(y/x)))(x = y)],$$

where  $y$  is free for  $x$  in  $\varphi$  (and therefore in  $trs'(\varphi)$ ). That is, by (MP4) and elementary transformations, the above instance of (CP) is equivalent to:

$$(\exists A)(\forall x)[(\exists yA)(x = y) \leftrightarrow (\exists y)(trs'(\varphi(y/x)) \wedge (x = y))],$$

which in turn, by identity logic, is equivalent to:

$$(\exists A)(\forall x)[(\exists yA)(x = y) \leftrightarrow trs'(\varphi)].$$

**Metatheorem 5:** If  $\varphi$  is a theorem of second-order monadic predicate logic, then  $trs'(\varphi)$  is a theorem of our present conceptualist logic.

Finally, by metatheorem 3, to show that our conceptualist logic is decidable, we need only show that the formulas of this logic in which no complex names occur are decidable. To show this we prove the following metatheorem by induction on these formulas.

**Metatheorem 6:** If  $\varphi$  is a formula of our conceptualist logic and no complex names occur in  $\varphi$ , then  $\vdash \varphi \leftrightarrow trs'(trs^*(\varphi))$ .

**Proof.** As noted, we prove this metatheorem by induction on the formulas of our conceptualist logic in which no complex names occur. The case for atomic formulas, which consist only of identities, is of course immediate; and for negations and conditionals, again the proof is immediate. Suppose the metatheorem holds for  $\varphi$ ; then again it follows immediately that it holds for  $(\forall A)\varphi$ . The only interesting case is for  $(\forall xA)\varphi$ . But, by definition of  $trs^*$ ,  $trs^*((\forall xA)\varphi) = (\forall x)[F_A(x) \rightarrow trs^*(\varphi)]$ , and therefore, by definition of  $trs'$ ,  $trs'(trs^*((\forall xA)\varphi)) = (\forall x)[(\exists yA)(x = y) \rightarrow trs'(trs^*(\varphi))]$ ; and therefore, by the inductive hypothesis and (MP1),  $\vdash (\forall xA)\varphi \leftrightarrow (\forall x)[(\exists yA)(x = y) \rightarrow trs'(trs^*(\varphi))]$ , which completes our proof by induction. ■

It follows, accordingly, that if  $\varphi$  is a formula of our conceptualist logic in which no complex names occur, then to decide whether or not  $\varphi$  is a theorem of this logic it suffices to decide whether or not  $trs^*(\varphi)$  is a theorem of second-order monadic predicate logic. If the latter is not a theorem of second-order monadic predicate logic, then, by metatheorem 4,  $\varphi$  is not a theorem of our conceptualist logic; and if  $trs^*(\varphi)$  is a theorem of second-order monadic predicate logic, then, by metatheorem 5,  $trs'(trs^*(\varphi))$  is a theorem of our conceptualist logic, and therefore, by metatheorem 6, so is  $\varphi$ .

Hence, by metatheorem 3, the decision problem for our conceptualist logic is reducible to that of second-order monadic predicate logic.

**Metatheorem 7:** Our present conceptualist logic is both consistent and decidable.

## References

- [Church 1956] Church, Alonzo, *Introduction to Mathematical Logic*, Princeton University Press, 1956.
- [Cocchiarella 1996] Cocchiarella, Nino B., “Conceptual Realism as a Formal Ontology,” in *Formal Ontology*, eds. R. Poli and P. Simons, Kluwer Academic Pub. Dordrecht, pp. 27–60, 1996.
- [Cocchiarella 2001] Cocchiarella, Nino B., “A logical reconstruction of medieval terminist logic in conceptual realism”, *Logical Analysis and History of Philosophy* 4 (2001): 35–72.
- [Geach 1980] *Reference and Generality*, third edition, Cornell University Press, Ithaca, 1980.
- [Henry 1972] Henry, D.P., *Medieval Logic and Metaphysics*, Hutchison University Library, London, 1972.
- [Iwanuś 1973] Iwanuś, B. “On Leśniewski’s Elementary Ontology,” *Studia Logica* xxxi (1973): 73-119. (Reprinted in Szrednicki *et al* 1984].)
- [Küng & Canty 1970] Küng, Guido and John T. Canty, “Substitutional quantification and Leśniewskian quantifiers”, *Theoria* xxxvi (1970): 165-182.
- [Lewjewski 1958] Lewjewski, Czesław, “On Leśniewski’s ontology”, *Ratio* 1, no.2 (1958): 150-176.

- [Slupecki 1955] J. Slupecki, "S. Leśniewski's Calculus of Names," *Studia Logica*, vol III (1955): 7-72. (Reprinted in Srzednicki *et al* 1984].)
- [Srzednicki *et al* 1984] J. Srzednicki, V.F. Rickey, and J. Czelakowski, eds., *Leśniewski's Systems, Ontology and Mereology*, Martinus Nijhoff Pub., The Hague, 1984.