A LITTLE MORE LIKE ENGLISH

John T. KEARNS

1. *Quantified Phrases*

Logical formal languages can be used as instruments for studying natural languages. If certain features of a natural language are incorporated in a formal language, then an understanding of these features in the formal language yields an understanding of them in the natural language. It is a great advantage of using formal languages over studying natural languages directly that formal languages can incorporate a few features at a time. The respects in which a formal language is unlike the intended natural language are generally respects in which the formal language is simpler and more perspicuous than the natural language. These make the formal language easier to study, but allow the formal language to constitute a partial theory of the natural language.

There are many syntactic and semantic respects in which customary first-order languages are unlike English. One respect to be considered in this paper concerns what I shall call *quantified noun phrases*, or, more simply, *quantified phrases*. These include such expressions as the following: every man, no woman, some child, a dog, an apple, the mayor of New York City. In English and (some) other natural languages, quantified phrases can occupy the same positions in a sentence that are occupied by proper nouns. For example,

(1) Tom sees Mary. (2) Tom sees a girl.

are both English sentences. In (2), a quantified phrase is found in a proper noun's position.

In first-order languages, (1) and (2) receive very different translations; they might come out this way: (3) $S(t, m)$ and (4) $(\exists x)\,[G(x)\ \&\ S(t, x)]$. It *has* been maintained that the difference between (3) and (4) brings out the difference between the

logical forms of two superficially similar English sentences. On such a view, a sentence's logical form is a mysterious entity lurking beneath the sentence's deceptive surface. I find that view implausible. The fact that there is a bigger difference between (3) and (4) than there is between (1) and (2) simply shows that in this respect the formal language is unlike English.

It has been argued (see [1]) that Lesniewski's system/language of Ontology is more like English (or Polish) than are first-order languages, because in Ontology the category of common nouns (allegedly) includes proper nouns. So in Ontology common nouns occupy the same positions in sentences as are occupied by proper nouns. However, this does not make Ontology more natural than first-order languages. For Ontology doesn't contain any proper nouns; the nearest thing to a proper noun is a common noun that denotes a single individual. Another difference between Ontology and English is that in Ontology the definite article is built-in to the «verb». An Ontological sentence '$A \varepsilon B$' means «The A is a B», but the 'ε' means «The...is a...»

In this paper, a language $\mathscr{L}$ will be developed which allows quantified phrases in the same places where individual constants can go. But $\mathscr{L}$ constitutes only a modest departure from first-order languages with identity. ($\mathscr{L}$ shares many features with the language developed by Montague in [2].)

2. *The basic syntax*

In developing $\mathscr{L}$, I will depart somewhat from standard terminology. In particular, I will not use the expression 'formula' (or 'well-formed formula') for labelling expressions obtained from sentences by substituting (free) individual variables for individual constants. I am doing this because I do not think bound variables «behave» like either proper nouns or (most) pronouns in English. They are simply «place holders», marking positions affected by quantifiers. (But in English there are *some* pronouns that do behave like bound variables; reflexive pronouns and possessive pronouns do this.)

The «sentential» expressions of $\mathscr{L}$ are *predicates*. The number of argument places in a predicate is the number of distinct free variables it contains. (A sentence is a O-adic predicate.) The order of the argument places in a predicate is (ordinarily) the order of the first free occurrences of the individual variables. If some predicate is written $\varphi(\alpha_1,\ldots, \alpha_n)$ for $n > 0$, then φ is an *n*-adic *predicate functor*.

$\mathscr{L}$ contains the following simple ingredients:

(1) Punctuation: (,), [,], the comma:,
(2) Individual constants (proper names): a, b, c, $a_1,\ldots$
(3) Individual variables: x, y, z, $x_1,\ldots$
(4) For every $n > 0$, *n*-adic predicate functors: F^n, G^n, H^n, $F^n_1,\ldots$
(5) Two logical predicate functors: $=$ (a binary functor), T (a monadic functor)
(6) Connectives: $\sim$, &, v, $\supset$
(7) Quantifiers: $\forall$, $\mathscr{A}$, ι
(8) Passive voice symbol: $\breve{}$
(9) Abstraction symbol: $\ni$

The connectives, quantifiers, passive voice symbol, and the abstraction symbol are the *operators* of $\mathscr{L}$. The predicate functors listed in (4) and (5) are *simple*. If φ is a simple *n*-adic predicate functor other than '=,' and $\alpha_1,\ldots, \alpha_n$ are distinct individual variables, then $\varphi(\alpha_1,\ldots, \alpha_n)$ is an *atomic predicate*. If α_1, α_2 are distinct individual variables, then $[\alpha_1 = \alpha_2]$ is an *atomic predicate*. (The predicate 'T(x)' means x *is a thing*—i.e., x is an individual in the domain.)

In describing the syntax of $\mathscr{L}$, I will use *transformations* to generate complex expressions from simpler ones. These transformations are not borrowed from any linguistic theory. A transformation is simply an effective function *from* certain expressions *to* an expression. The semantics of $\mathscr{L}$ will be described concurrently with the syntax. Given the set (or truth value) denoted by a predicate A, where A' is obtained from A by a transformation, I explain how the value of A' is determined for this transformation.

$\mathscr{L}$ is interpreted in nonempty domains. Let $\mathscr{D}$ be nonempty. Then an *interpreting function of* $\mathscr{L}$ *for* $\mathscr{D}$ is a function f defined on the individual constants and simple predicate functors of $\mathscr{L}$

which assigns an individual in $\mathscr{D}$ to each individual constant, and which assigns a set of n-tuples of individuals of $\mathscr{D}$ to each n-adic predicate functor. An interpreting function must assign $\{\alpha \mid \alpha \varepsilon \mathscr{D}\} = \mathscr{D}$ to 'T' and $\{\langle \alpha, \alpha \rangle \mid \alpha \varepsilon \mathscr{D}\}$ to '='.

Let $\mathscr{D}$ be nonempty and let f be an interpreting function of $\mathscr{L}$ for $\mathscr{D}$. Let $\varphi(\alpha_1 \ldots, \alpha_n)$ be an atomic predicate of $\mathscr{L}$, and let $f(\varphi)$ V. Then the *value of* $\varphi(_1, \ldots, \alpha_n)$ *for* f is V; we can also write this: $f[\varphi(\alpha_1, \ldots, \alpha_n)] = V$. If $[\alpha_1 = \alpha_2]$ is an atomic predicate, then $f[\alpha_1 = \alpha_2] = \{\langle \beta, \beta \rangle \mid \beta \varepsilon \mathscr{D}\}$.

In describing transormations, I will first explain the syntactic aspect of a transformation. Then I will give an informal account of the semantic effect of the transformation. Following this, *for some transformations,* will be a precise and careful statement of the semantic effect of the transformation. The careful statement will be italicized. (For those transformations where the precise statement is omitted, the diligent reader can easily supply the omission.)

(1) The *reflexive* transformation transforms a predicate A into A' by replacing free occurrences of a variable α by occurrences of a different variable which also occurs free in A. For example, the atomic predicate '[x = y]' is transformed into '[x = x]' by the reflexive transformation. In this transformation, a free variable cannot be replaced if it occurs within the scope of an operator binding occurrences of the new variable. The significance of this transformation should be clear; it takes a predicate like 'x hit y' into 'x hit x-self.'

In talking about the distinct individual variables occurring free in a predicate, I will speak of their *canonical* order. This is normally the order of the first free occurrence of each variable (the exceptions to this normal order are explained with the quantifier transformations). For transformations (1)-(6), the initial predicate will be A, and the result of the transformation is A'. The distinct individual variables occurring free in A, in canonical order, are $\alpha_1, \ldots, \alpha_n$. The value of A for f is V.

Let A' be obtained by the reflexive transformation from A by replacing occurrences of α_i by α_j $(i \neq j)$. Then $f(A') = \{\langle d_1, \ldots, d_{n-1} \rangle \mid \langle d_1, \ldots, d_{i-1}, d_j, \ldots, d_{n-1} \rangle \varepsilon V\}$.

(2) The *constant introduction* transformation replaces the free occurrences of a variable α_1 by an individual constant β. *Let* $f(\beta) = e$. *Then* $f(A') =$ $\{\langle d_1,\ldots, d_{n-1}\rangle \mid \langle d_1,\ldots, d_{i-1}, e,\ldots, d_{n-1}\rangle \in V\}$. *If* $n = 1$, *then* $f(A') = t$(*ruth*) *iff* $e \in V$.

The quantifiers of $\mathscr{L}$ are '$\forall$' (every), '$\mathscr{A}$' (a, an, some), ι (the); these are the *universal, indefinite,* and *definite* quantifiers, respectively. For each quantifier, there are two kinds of quantified phrase. If φ is a monadic predicate functor and α is an individual variable, then $\varphi\alpha$ is an *indexed monadic predicate functor.* Let ψ be either a monadic predicate functor or an indexed monadic predicate functor. Then (i) $\forall\psi$ is a *universally* quantified phrase, (ii) $\mathscr{A}\psi$ is an *indefinitely* quantified phrase, and (iii) $\iota\psi$ is a *definitely* quantified phrase. For each kind of quantified phrase, the transformation which consists in applying that phrase to a predicate is subdivided into three transformations. These are exactly parallel for each quantifier, so I will explain them in detail only for the universal quantifier.

(3) Let φ be a monadic predicate functor which contains no free individual variables in common with A.

(3a) Then $\forall\varphi\alpha_i A(= A')$ is obtained by the universal transformation.

(3b) If the first free occurrence of α_i in A is within the scope of no operators in A, then A' results by replacing the first free occurrence of α_i in A by $\forall\varphi\alpha_i$.

(3c) If there is only one free occurrence of α_i in A, and this occurrence is within the scope of no operator in A, then A' results by replacing the free occurrence of α_i by $\forall\varphi$.

The three universal transformations yield equivalent predicates. To illustrate these transformations, I will use capital letters as if they were predicate functors of $\mathscr{L}$. Suppose 'P(x)' means x *is a person,* and 'M(x)' means x *is mortal.* Then 'Everyone is mortal' can be translated by: $\forall P$x M(x)—for every person x, x is mortal. But we can also say the same thing by writing '$M(\forall P$x)'—every person x is mortal. Or (better) '$M(\forall P)$' —every person is mortal. If 'L(x, y)' means x *loves* y, then '$L(\forall P, \forall P)$' is «every person loves every person.' And '$L(\forall P$y, y)'

is «every person y loves y»—i.e., every person loves himself/herself. (If we dropped the index in the quantified phrase of this last example, we get '$L(\forall P, y)$'; this is a monadic predicate rather than a sentence—it denotes whatever is loved by everyone.)

The prefixed quantified phrase of a sentence can be «moved inside» the sentence only if the internal position is not within the scope of an operator. The sentence '$\forall Px \sim M(x)$' is *not* equivalent to '$\sim M(\forall P)$,' because the single free occurrence of 'x' in '$\sim M(x)$' is within the scope of the operator '$\sim$.' We cannot obtain '$\sim M(\forall P)$' from '$\sim M(x)$' by the quantifier transformation. But we can obtain this sentence by starting with '$M(x)$,' obtaining '$M(\forall P)$,' and then negating that sentence. (The sentence '$\sim M(\forall P)$' is equivalent to '$\sim \forall Px\, M(x)$' rather than to '$\forall Px \sim M(x)$.') In $\mathscr{L}$ the scope of an operator cannot always be a complete expression. Instead we determine when an argument place is *within the scope* of an operator. When the quantified phrase is in front, as in $\forall\varphi\alpha\, A$, then every argument place in A is within the scope of the quantifier. But when the quantified phrase is moved inside A, yielding A', then only argument places to the right of the quantified phrase are within the scope of the quantifier.

The *canonical order* of the argument places in a predicate is ordinarily the order of the first free occurrence of the distinct free variables in the predicate. But monadic predicate functors may contain free occurrences of individual variables. When this happens, a transformation yielding a prefixed quantifier could give a different order to the free individual variables than an equivalent transformation yielding an internal quantifier. A predicate with a prefixed quantifier will be canonical for the quantifier transformations. This predicate is used to determine the order of the argument places. The canonical order of distinct free individual variables in a predicate A is the order of the first free occurrences of these variables in the corresponding predicate B in which all quantifiers are in canonical position.

A universally quantified phrase is understood to have existential force. In case there aren't any φ's, then $\forall\varphi\alpha\, A$ is false.

Let A' be obtained from A by the universal transformation, either by prefixing $\forall\varphi\alpha_i$ *to A, by inserting* $\forall\varphi\alpha_i$ *in place of the first free occurrence of* α_i *in A, or by inserting* $\forall\varphi$ *in place of the single free occurrence of* α_i *in A. Let the distinct individual variables occurring free in* φ, *in canonical order, be* $\beta_1,\ldots,\beta_m$. *Let* $f(A) = V$, $f[\varphi(\alpha_i)] = X$. *Then the value of A' for f is* $\{\langle d_1,\ldots, d_m, e_1,\ldots, e_{n-1}\rangle \mid$ *there is an* $r \in \mathscr{D}$ such that $\langle d_1,\ldots, d_m, r\rangle \in X$ *and, for every such r,* $\langle e_1,\ldots, e_{i-1}, r, \ldots, e_{n-1}\rangle \in V\}$. *In case* $m = 0$, $n = 1$, $f(A') = t$ *iff there is an* $r \in \mathscr{D}$ *such that* $r \in X$ *and, for every* $r \in X$, we have $r \in V$.

(4) The indefinite quantifier can be read «a,» «an,» or «some.» So '$\mathscr{A}Px\,L(x, x)$' is «For some person x, x *loves* x.» And '$L(\forall P, \mathscr{A}P)$' is «Every person loves some person.» The indefinite quantifier is used to say that at least one individual of the kind indicated has (or does) etc.

(5) The definite quantifier is used to say that the one and only individual of the indicated kind is etc. So '$M(\iota P)$' is «The (sole) person is mortal.» And '$[\mathscr{A}P = \iota P]$' *is* «Someone is the sole person.»

(6) The *negation* transformation takes A into $\sim A$. The positions in A are within the scope of '$\sim$' *The value of* $\sim A$ *for f is* $\{\langle d_1,\ldots, d_n\rangle \mid \langle d_1,\ldots, d_n\rangle \notin V\}$. *If* $n = 0$, *then* $f(\sim A) = t$ *iff* $f(A) = f$.

(7)-(9) If A is an n-adic predicate, and B is an m-adic predicate which has no free variables in common with A, then these transformations yield $[A \,\&\, B]$, $[A \vee B]$, $[A \supset B]$. The positions in A and B are within the scope of the binary operator. These connectives have their ordinary (logical) meanings.

The sentence '$\forall Px\,\mathscr{A}Py\,L(y, x)$' is used to say that everyone is loved by someone. This sentence is not equivalent to a sentence with both quantified phrases moved inside. The sentence *is equivalent to* '$\forall Px\,L(\mathscr{A}P, x)$.' The prefixed quantified phrase cannot be moved inside, for the free occurrence of 'x' in '$L(\mathscr{A}P, x)$' is within the scope of '$\mathscr{A}$' In English, this problem is

solved by using the passive voice. We will incorporate that solution in $\mathscr{L}$.

(10) If ψ is a binary predicate functor and α_1, α_2 are distinct individual variables, then the *passive voice* transformation takes $\psi(\alpha_1, \alpha_2)$ into $\breve{\psi}(\alpha_1, \alpha_2)$.

If $f[\psi(\alpha_1, \alpha_2)] = \{\langle d_1, d_2\rangle \mid \ldots\} = V$, *then* $f[\breve{\psi}(\alpha_1, \alpha_2)] = \{\langle d_1, d_2\rangle \mid \langle d_2, d_1\rangle \varepsilon V\}$.

(11) Let φ be a monadic predicate functor, containing free occurrences of distinct individual variables $\beta_1, \ldots, \beta_m$ and no others (this is their canonical order). Let A be a predicate which contains no free individual variables in common with φ, and which contains free occurrences of distinct individual variables $\alpha_1, \ldots, \alpha_n$ and no others (this is their canonical order). Let γ be an individual variable which is either α_1 or which does not occur free in either φ or A. Then the *relative clause* transformation yields the predicate $[\varphi\alpha_i \ni A](\gamma)$. This transformation is also called the *abstraction* transormation. The predicate $[\varphi\alpha_i \ni A](\gamma)$ is read «γ is a $\varphi\alpha_i$ such that A.» For example, '$[Px \ni L(x, \mathscr{A}P)](a)$' is «$a$ is a person x such that x loves someone» —i.e., α is a person who loves someone. The occurrence of α_i in $\varphi\alpha_i$ and the free occurrences of α_i in A are bound occurrences in $[\varphi\alpha_i \ni A]$. The positions in $[\varphi\alpha_i \ni A]$ are within the scope of the displayed occurrence of '$\ni$.'

The relative clause transformation makes it possible to construct complex functors which can occur in quantified phrases. Suppose 'α' names Anne and '$K(x, y)$' means x *knows* y. Then '$L(\forall[Px \ni K(x, a)], a)$' claims that everyone who knows Anne loves Anne.

Let $f[\varphi(\gamma)] = X$ *and* $f(A) = V$.

Then $f([\varphi\alpha_i \ni A](\gamma)) =$
$\{\langle d_1, \ldots, d_m, e_1, \ldots, e_{n-1}, r\rangle \mid \langle d_1, \ldots, d_m, r\rangle \varepsilon X$ *and* $\langle e_1, \ldots, e_{i-1}, r, \ldots, e_{n-1}\rangle \varepsilon V\}$.

3. *The first-order fragment of* $\mathscr{L}$

In first-order languages, an English sentence of the form 'Every F is G' is translated '$(\forall x)[F(x) \supset G(x)]$.' There are (at least) two respects in which the truth-conditions of the translation diverge from those of the original: (i) the translation is vacuously true if there are no F's and (ii) everything in the domain is an instance of the quantified translation. In $\mathscr{L}$, the English sentence can be translated '$G(\forall F)$.' The translation in $\mathscr{L}$ does not admit of vacuous truth, and only the F's are its instances. The $\mathscr{L}$ translation is also syntactically closer to the English sentence than is the first-order translation, because the $\mathscr{L}$ translation does not contain a binary connective.

While $\mathscr{L}$ provides translations of English sentences that are syntactically superior to the first-order translations, $\mathscr{L}$ also «incorporates» first-order sentences. The quantified phrase '$(\forall x)$' of a first-order language has the same significance as '$\forall T x$' in $\mathscr{L}$. And '$(\exists x)$' amounts to '$\mathscr{A} T x$.' The requirement that a first-order language be interpreted in a nonempty domain corresponds to the requirement that $\forall T x\ A$ not admit of vacuous truth. (But it is easy to change the semantics of a first-order language to accommodate the empty domain; it is equally easy to change the semantics of $\mathscr{L}$ so that the universal quantifier admits of vacuous truth.) Even without vacuous truth for '$\forall T x$,' a sentence '$\forall T x\ [F(x) \supset G(x)]$' *is* true if there are no F's.

The *first-order fragment* is the sublanguage of $\mathscr{L}$ which (1) contains those predicates in which the only quantifiers are '$\forall$' and '$\mathscr{A}$,' where these occur in quantified phrases which are in canonical position, which phrases all contain (only) the predicate functor 'T,' and (2) contains no predicates obtained by the passive voice transformation or the relative clause transformation. An ordinary first-order language with identity can be regarded as an abbreviated version of the first-order fragment of $\mathscr{L}$. In the abbreviated version, occurrences of 'T' in quantified phrases are omitted, and '$\mathscr{A}$' is replaced by '$\exists$.' It is an easy matter to prove that each sentence (and predicate) in $\mathscr{L}$ is equivalent to a sentence (predicate) in the first-order fragment

of $\mathscr{L}$ (and so to a sentence in any first-order language with identity). A first-order language is semantically as powerful as $\mathscr{L}$, but a first-order language cannot accommodate the syntactic features of $\mathscr{L}$.

4. *Identity*

The symbol '=' in $\mathscr{L}$ is a counterpart to the 'is' of identity in English. This 'is' is not very often used to connect two proper names in English sentences, except in such philosophers' examples as 'Cicero is Tully.' However, the 'is' of identity is very useful when quantifiers are present, and $\mathscr{L}$'s '=' illustrates how this 'is' works. For any sentence $\varphi(\alpha)$, where α is an individual constant, is equivalent to $[\alpha = \mathscr{A}\varphi]$. So given the indefinite quantifier and the identity sign, it is never necessary to use a monadic predicate functor in a sentence of the form $\varphi(\alpha)$. And in English, many common nouns are not closely related to intransitive verbs. Since we can say «Tom is a carpenter,» we have no need to say «Tom carpents.»

Frege claimed that the 'is' of identity is quite different from the 'is' of predication. But the language $\mathscr{L}$ shows how the 'is' of identity is used to make predications without changing its meaning. To insist that English contains two 'is's is to take standard first-order languages with identity as «capturing» the logical forms of the English sentences they translate. But first-order languages only provide *semantic* approximations to such sentences as 'Arnold is a carpenter,' 'Some carpenter is a woodworker,' and 'Every carpenter is a woodworker.' If '*a*' names Arnold, '$C(x)$' means x *is a carpenter,* and '$W(x)$' means x *is a woodworker,* then $\mathscr{L}$ provides translations that are syntactically *and* semantically appropriate in the following: $[a = \mathscr{A}C]$, $[\mathscr{A}C = \mathscr{A}W]$, and $[\forall C = \mathscr{A}W]$.

5. *Deduction*

The natural deduction system $\mathscr{S}$ employs tree proofs in which only sentences can occur as steps.

The rules for connectives are entirely familiar.

& INTRODUCTION

$$\frac{A \quad B}{[A \ \& \ B]}$$

& ELIMINATION

$$\frac{[A \ \& \ B]}{A} \qquad \frac{[A \ \& \ B]}{B}$$

$\vee$ INTRODUCTION

$$\frac{A}{[A \vee B]} \qquad \frac{B}{[A \vee B]}$$

$\vee$ ELIMINATION

$$\frac{[A \vee B] \quad \overset{\{A\}}{C} \quad \overset{\{B\}}{C}}{C}$$

Occurrences of the premisses in braces which are over the indicated occurrences of C are cancelled by this rule.

$\supset$ INTRODUCTION

$$\frac{\begin{matrix}\{A\}\\ B\end{matrix}}{[A \supset B]}$$

$\supset$ ELIMINATION

$$\frac{A \quad [A \supset B]}{B}$$

CONTRADICTION ELIMINATION

$$\frac{A \quad \sim A}{B}$$

$\sim$ ELIMINATION

$$\frac{\begin{matrix}\{\sim A\}\\ A\end{matrix}}{A}$$

In writing the rules for quantifiers, it is understood that φ is a monadic predicate functor, and that α is an individual variable. All three quantifiers have similar INSERTION and EXTRACTION rules. Let Q be '$\forall$,' '$\exists$,' or 'ι.'

Q INSERTION

$$\frac{Q\varphi\alpha\ A}{A'}$$

α occurs free in A. The first free occurrence of α in A is within the scope of no operators in A. A' is obtained from A by replacing the first free occurrence of α in A by $Q\varphi\alpha$.

$$\frac{Q\varphi\alpha\ A}{A'}$$

α has a single free occurrence in A. This free occurrence is within the scope of no operators in A. A' is obtained from A by replacing the free occurrence of α in A by $Q\varphi$.

Q EXTRACTION

These are obtained by «turning over» the Insertion rules. The conditions are the same.

$\mathscr{A}$ INTRODUCTION

$$\frac{\varphi(\beta) \quad \S\,{}^{\alpha}_{\beta} A\,|}{\mathscr{A}\varphi\alpha\ A}$$

α occurs free in A. β is an individual constant. The expression $\S\,{}^{\alpha}_{\beta} A\,|$ denotes the result of replacing *free* occurrences of α in A by β.

$\mathscr{A}$ ELIMINATION

$$\frac{\mathscr{A}\varphi\alpha\ A \qquad \{\S\,{}^{\alpha}_{\beta} A\,| \ \&\ \varphi(\beta)\}\ C}{C}$$

β is an individual constant which does not occur in $\mathscr{A}\varphi\alpha\ A$, in C, or in any uncancelled hypothesis (other than the one in braces) of the proof leading to the occurrence of C on the line.

SUBALTERNATION $\forall$ ELIMINATION

$$\frac{\forall\varphi\alpha\ A}{\mathscr{A}\varphi\alpha\ A} \qquad \frac{\forall\varphi\alpha A \quad \varphi(\beta)}{\S\,{}^{\alpha}_{\beta} A\,|}$$

β is an individual constant.

∀ INTRODUCTION

$$\frac{\varphi(\mathscr{A}\varphi) \qquad \overset{\{\varphi(\beta)\}}{A}}{\forall\varphi\alpha\, S^{\beta}_{\alpha} A \mid}$$

β is an individual constant which does not occur in φ. β does not occur in any uncancelled hypothesis except φ(β) in the proof leading to *A*. β occurs in *A* but not within the scope of an operator binding α.

= INTRODUCTION

For every individual constant β, the sentence [β = β] can be introduced as a conclusion from no premisses (as an axiom).

= ELIMINATION

$$\frac{[\beta = \gamma] \qquad A}{A'}$$

A' results from *A* by replacing some occurrences of the constant β by occurrences of the constant γ.

¬ INTRODUCTION

$$\frac{\forall\varphi\gamma\, [\gamma = \beta] \; S^{\alpha}_{\beta} A]}{\neg\varphi\alpha\, A}$$

α occurs free in *A*. β is a constant.

¬ ELIMINATION

$$\frac{\neg\varphi\alpha\, A \qquad \overset{\{\forall\varphi\gamma\, [\gamma = \beta] \,\&\, S^{\alpha}_{\beta} A \mid\}}{C}}{C}$$

βis an individual constant which does not occur in ¬φα *A*, in *C*, or in any uncancelled hypothesis (other than the one in braces) in the proof leading to the occurrence of *C* on the line.

$\breve{}$ INTRODUCTION $\breve{}$ ELIMINATION

$$\frac{\varphi(\alpha, \beta)}{\breve{\varphi}(\beta, \alpha)} \qquad \frac{\breve{\varphi}(\alpha, \beta)}{\varphi(\beta, \alpha)}$$

α and β are individual constants.

$\ni$ INTRODUCTION $\ni$ ELIMINATION

$$\frac{[\varphi(\beta) \;\&\; \S\,{}^{\alpha}_{\beta} A\,|]}{[\varphi\alpha \ni A](\beta)} \qquad \frac{[\varphi\alpha \ni A](\beta)}{[\varphi(\beta) \;\&\; \S\,{}^{\alpha}_{\beta} A\,|]}$$

α occurs free in A. β is an individual constant.

T INTRODUCTION

If β is an individual constant, then $T(\beta)$ can be introduced as a conclusion from no premisses.

If B is the conclusion of a tree proof whose uncancelled hypotheses are included among $A_1, \ldots, A_n$, then $A_1, \ldots, A_n/B$ is a *sequence theorem* of $\mathscr{S}$. If $/B$ is a sequence theorem of $\mathscr{S}$, then B is a *sentence theorem* of $\mathscr{S}$. The system $\mathscr{S}$ is sound, and it is complete with respect to logically true sentences, with respect to valid (finite) inference sequences, and with respect to the logical consequences of sets of sentences. These results are proved by slightly modifying similar proofs for first-order systems.

6. *Some conclusions*

The language $\mathscr{L}$ is an improvement over first-order languages for the purposes of gaining an understanding of and providing an explanation for some grammatical and logical features of English. However, it is important to realize that $\mathscr{L}$ does not improve on first-order languages with respect to what can be said. Every sentence of $\mathscr{L}$ is equivalent to a sen-

tence of the first-order fragment of $\mathscr{L}$. Conversely, $\mathscr{L}$ can be obtained as the definitional extension of a first-order language. But this last fact does not confer a privileged status on first-order languages. It is (or should be) obvious by this time that it is an arbitrary matter which expressions in a formal language are taken as primitive and which are defined. The fact that an adequate formal language can be developed from a small set of primitive symbols tells us next to nothing about the «ultimate units» of thought, language, or reality.

The virtues of $\mathscr{L}$ belong to $\mathscr{L}$ whether $\mathscr{L}$ is adopted as a primitive language or is obtained as the definitional extension of a first-order language. These virtues primarily involve the light that $\mathscr{L}$ sheds on the connection between the syntactical constructions and logical features of English sentences. Consider what happens when an English sentence is translated in a first-order language. If the translation is not syntactically similar to the English sentence, then the translation is justified solely by our insight into the meanings or truth conditions of the two sentences. When a translation in $\mathscr{L}$ is provided which is syntactically similar to the English sentence, the need to appeal to insight is lessened. Once the English sentence is translated in $\mathscr{L}$, we can move from this translation to the sentence of $\mathscr{L}$ which corresponds to the first-order translation. But now this move can be recognized for the inference that it is. And $\mathscr{L}$ provides the resources for analyzing this inference.

The behavior of quantified phrases in $\mathscr{L}$ approximates their behavior in English. This is because $\mathscr{L}$ possesses the variety of quantified phrases made possible by incorporating monadic functors into these phrases, and $\mathscr{L}$ permits quantified phrases in the slots for proper names. The passive voice and the relative clause transformations make clear the importance of the analogous English «transformations» with respect to quantifiers. And the identity sign in $\mathscr{L}$ illustrates and explains how the 'is' of identity also serves for predication. The syntactic features and the logical expressions of $\mathscr{L}$ have roughly the same semantic force as their English counterparts. Both syntactically and semantically, $\mathscr{L}$ provides a model of a portion of English. The language $\mathscr{L}$ together with the system $\mathscr{S}$ consti-

tutes a partial theory of the logical features of English. It is an explanatory theory because it makes clear the connection between syntactical form and logical force.

Department of Philosophy,
State University of New York at Buffalo
John T. KEARNS

REFERENCES

[1] E. C. LUSCHEL, *The Logical Systems of Lesniewski*, North-Holland Publishing Company, Amsterdam, 1962.

[2] Richard MONTAGUE, «English as a Formal Language,» in *Formal Philosophy, Selected Papers of Richard Montague*, edited by R. H. Thomason, Yale University Press, New Haven, 1974, 188-221.