TERM CALCULUS AND A GENERAL FORMAL LANGUAGE

A Thesis

Presented to the

Faculty of

San Diego State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Arts

in

Mathematics

by

Sandor M. Turke

Fall 2016

SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the

Thesis of Sandor M. Turke:

TERM CALCULUS AND A GENERAL FORMAL LANGUAGE

J. Carmelo Interlando, Chair
Department of Mathematics and Statistics

Vadim Ponomarenko
Department of Mathematics and Statistics

Carl Eckberg
Department of Computer Science

DECEMBER 19, 2016

Approval Date

DEDICATION

To my mother and father and to my wife Charity.

Pure mathematics is the poetry of logical ideas.

– Albert Einstein

ABSTRACT OF THE THESIS

TERM CALCULUS AND A GENERAL FORMAL LANGUAGE
by
Sandor Turke
Master of Arts in Mathematics
San Diego State University, 2016

This thesis is a contribution to mathematical logic. In the language of predicate calculus, I change the syntactic structure of the atomic formula from a predicate relation on terms, where terms represent individuals, to a verb as a relation on nouns, where nouns represent either individuals or properties. Based on the verb on nouns syntactic structure for atomic formulas, I construct a formal language that has all the properties of a first-order predicate calculus and, furthermore, that can model natural syntax, the syntax of natural languages.

While most mathematicians and logicians acknowledge the success of predicate logic for mathematics, some logicians have criticized predicate logic as being unnatural in a sense, in that its syntax cannot adequately model natural syntax. Logicians who refute predicate logic as a logic for natural languages maintain that term logic is the logic of natural languages and models how we actually reason. I show that the constructed formal language is mathematically rigorous enough to be a basis for the language of predicate calculus by constructing a predicate calculus from the formal language. I also show that the formal language is natural enough to model the syntax of term logic by constructing a formal calculus for term logic from the formal language.

Theoretical implications of the formal language are found in the fields of mathematics and linguistics. I show how the formal language may be an approach and mathematical framework for the foundations of mathematics and language in general.

TABLE OF CONTENTS

LIST OF TABLES

GLOSSARY

— the copula verb symbol which means ’is’.

$\exists$ the existential quantifier symbol which means ’there exists’ or ’some’.

$\forall$ the universal quantifier symbol which means ’for all’ or ’all’.

$\wedge$ the conjunction symbol which means ’and’.

$\vee$ the disjunction symbol which means ’or’.

$\models$ Logical implication: $\Gamma \models \phi$ means the set of formulas Γ *logically implies* the formula ϕ.

$\models_{\mathfrak{A}}$ Model symbol: $\models_{\mathfrak{A}} \phi[s]$ means the interpretation $\mathfrak{A}$ *satisfies* formula ϕ with object assignment s; and $\models_{\mathfrak{A}} \sigma$ means the sentence σ is *true* in interpretation $\mathfrak{A}$, or $\mathfrak{A}$ is a *model* of σ.

$\not-$ the negative copula verb symbol which means ’is not’.

$\rightarrow$ the implication symbol which means ’implies’.

$\sim$ the negation symbol which means ’not’, when applied to a formula, or ’un’, when applied to a term.

$\vdash$ Deduction symbol: $\Gamma \vdash \phi$ means the set of formulas Γ *deduces* formula ϕ (ϕ is *deducible* from Γ), or ϕ is a *theorem* of Γ.

N the universe of the general formal language and the set of individual entities.

V the set of relations of the universe for the general formal language and the set of verbs in the metalanguage.

$\mathcal{L}_{\mathcal{P}}$ A formal language with atomic formulas that has a the predicate on terms syntactic structure.

$\mathcal{L}_{\mathcal{T}}$ A formal language for term logic that is based on the general formal language.

$\mathcal{L}_{\mathcal{V}}$ A formal language with atomic formulas that has a verb on nouns syntactic structure; a.k.a the general formal language.

$\mathfrak{G}$ the interpretation, or structure, of the general formal language.

$\mathscr{N}$ the set of subsets of the universe of the general formal language and the set of properties.

$\mathsf{K_P}$ Formal theory with $\mathcal{L}_{\mathcal{P}}$ as its underlying language.

$\mathsf{K_V}$ Formal theory with $\mathcal{L}_{\mathcal{V}}$ as its underlying language.

S Set theory with $\mathcal{L}_{\mathcal{T}}$ as its underlying language.

T Formal theory with $\mathcal{L}_{\mathcal{T}}$ as its underlying language.

CC Classical Calculus: Sommers calculus of terms.

DG Dependency Grammar.

GG Generative Grammar.

MG Mathematical Grammar: the general formal language with a grammar system.

PC Predicate Calculus: A formal language with predicate on terms atomic formulas, like $\mathcal{L}_{\mathcal{P}}$, with logical axioms and no proper axioms.

RG Relational Grammar.

TC Term Calculus: A formal language $\mathcal{L}_{\mathcal{T}}$ with logical axioms and no proper axioms.

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my advisor, Professor Carmelo Interlando, for his encouragement and support in me to explore this thesis, even though the region of exploration was outside his area of specialty. His patient listening, insight, and confronting questions, guided me in the writing of this thesis.

I would also like to thank Professor Vadim Ponomarenko. The door to Professor Ponomarenko's office was always open whenever I had a question about mathematics, research or anything relevant to graduate life.

I would also like to thank Dr. Carl Eckberg of the computer science department for being in my thesis committee.

Finally, I would like to express my very profound gratitude to my family: my parents, my sisters and brothers, especially Richard and Tim, and my wife for encouraging me with unwavering belief in me and for supporting me spiritually throughout my years of study and through my research and writing this thesis. My accomplishment would not have been possible without them.

CHAPTER 1
INTRODUCTION

1.1 HISTORY

Aristotle's term logic with some relatively minor contributions of other ancient and medieval philosophers dominated western logic for millennia. Immanuel Kant considered Aristotles logic as a complete science, believing that there remains no significant discovery for intellectual explorers who wish to venture in the field of logic. A revolution in logic however happened. In the mid nineteenth century through the early twentieth century, logic was developed into a rigorous formal (mathematical) language which became known as predicate logic. Predicate logic was first discovered by Gottlob Frege and then systematically established in Whitehead's and Russell's *Principia Mathematica*, and it grew to become the first serious advancement and departure from the long reign of the term logic of Aristotle. Predicate logic developed to become the standard logic for not only philosophy and logic but also for linguistics and mathematics. It is the underlying language of set theory, a theory in which all of mathematics is described. Predicate logic is therefore the language of mathematics and is fundamental in the study of the foundations of mathematics. (For a history of predicate logic, I refer the interested reader to Jose Ferreiros [4].)

Predicate logic is not free of criticism from contemporary logicians and philosophers. These logicians may concede that the language of predicate logic works well for mathematics. However they contend that predicate logic does not adequately model natural syntax. For example, the sentence 'All men are animals' is mathematically modeled as a compound implication statement with variables: $\forall x\ Mx \rightarrow Ax$, which is read as 'for all x, x is a man implies x is an animal'. Yet, as we see, the sentence 'All men are animals' is a simple statement and variable free. Because predicate logic does not follow the syntax of natural languages, logicians further contend that, consequently, the deductive methods of predicate logic does not follow and represent how we naturally reason (See Sommers's *The Logic of Natural Language*[17]). These contentions began the revival of term logic in the 1970s. Term logic, being the logic based on natural syntax, is now considered to be the logic for natural languages and models how we naturally reason in everyday life.

1.2 Purpose

My purpose is to mathematically construct a general formal language that can be used as a basis for the mathematical language of predicate logic and for the natural language of term logic. By *formal language* I mean a language that has a set of symbols from which the formulas of the language are formed by explicit rules of formation; i.e. a formal language is the set of formulas formed out of symbols by rules of formation. I intend to show that the general formal language can be equipped with the deductive apparatus of a first-order predicate calculus and, furthermore, to provide a formal calculus for term logic.

This thesis provides a first mathematical construction for term logic that is similar to the mathematical construction of predicate logic. This shows that the general formal language is a mathematical language that can model the language of term logic and thus natural syntax. This thesis, therefore, provides a general formal language that is mathematically precise enough for a predicate calculus yet natural enough for a term calculus. It therefore acts as a bridge for mathematical languages and natural languages.

The general formal language, being a basis for natural language and the language of mathematics, has theoretical implications in the fields of linguistics as well as mathematics. In this thesis, I briefly explore the general formal language as a mathematical basis for current theories of grammar that purport to model universal grammar. The general formal language therefore may serve as a mathematical basis for universal grammar, which is the study of principles of grammar universal to all natural languages. In mathematics, I apply the general formal language as the underlying language of a set theory, instead of the language of predicate calculus which is set theory's usual underlying language. I discuss how this may shed light on the foundations of mathematics, since set theory is commonly applied as a foundational system for mathematics.

1.3 Overview

In Chapter 2, *Brief Presentation of Predicate Calculus*, I give a brief presentation of the construction of a predicate calculus. Readers who are familiar with the mathematical construction of predicate calculus may skip this chapter. I do refer to this chapter in later chapters.

In Chapter 3, *The Grammar of an Atomic Formula*, I argue that a verb on nouns is a more natural syntactic structure of the atomic formula than a predicate on terms. We explore choice examples of simple sentences in our natural language of choice, English, which expose the differences of the two syntactic structures in how well they accord with the semantic structure of the sentences. We will observe that a verb on nouns syntactic structure accords well with the semantics with a one to one correspondence whereas a predicate on terms is remote from the semantics.

In Chapter 4, *The General Formal Language*, I take noun symbols and verb symbols as primitive objects and proceed to construct a formal language that is based on the idea that a verb on nouns is the syntactic structure of the atomic formula. The construction of the formal language is similar to the construction of the syntax and semantics of a predicate calculus, yet with necessary modifications and added features.

I then provide a method of abbreviating the atomic formulas of the formal language from verb on nouns syntactic structures to predicate on terms syntactic structures, which will show that the constructed formal language may include the formal language of a predicate calculus and therefore inherent the deductive system of a predicate calculus and its properties. This makes the formal language with a verb on nouns syntactic structure for atomic formulas a general formal language.

In Chapter 5, *Term Calculus*, I show that the constructed formal language can model natural syntax by furthering its construction to a formal language that models the language of term logic, the logic of natural syntax. I then proceed to construct a formal calculus for the formal language of term logic by providing inference rules based on the relation of terms. Hence the constructed formal calculus is a calculus of terms which I call term calculus.

I also show that the formal language of term calculus can apply to existing term logics. Specifically, I show that the language of term calculus can establish Jan Lukasiewicz's axiomatic method of Aristotle's term logic as a formal calculus. I moreover provide a one to one correspondence between the formal language of term calculus with the language for Sommers calculus of terms, which is an algebraic calculus for term logic. This shows that the general formal language may serve as a mathematical language for both predicate logic and term logic. This further shows that the language of term calculus may include an algebraic as well as axiomatic calculi.

In Chapter 6, *Theoretical Implications of the General Formal Language*, I suggest how we may apply the general formal language specifically in mathematics and linguistics and discus the theoretical ramifications.

For mathematics, I provide the beginnings of a set theory with the language of term calculus as its underlying language. (A predicate calculus is the usual underlying language of set theory). In this set theory, I derive the fundamental relations of set theory, namely equality, membership, inclusion, and intersection from a single relation of the language of term calculus.

For linguistics, I provide the beginnings of a grammar system for the general formal language. The idea of the verb as a relation exposes a deeper grammar for the atomic formula.

A grammar system, for the general formal language, is a system of definitions and rules that form all the kinds of grammatical sentences in a natural language and rules that explain grammatical phenomenon.

The general formal language with a grammar system I call mathematical grammar. I make suggestions on how to develop the grammar system and thus mathematical grammar. I suggest how mathematical grammar may serve as a mathematical basis for current theories of grammar, specifically for dependency grammars, generative grammar, and relational grammar. This then leads to the consideration of mathematical grammar as a mathematical basis for universal grammar, the theory of the foundations of natural language.

Finally, in the *Conclusion*, I summarize what has been established and conclude by advocating for the general formal language as a mathematical framework and approach for the foundations of both natural languages and mathematical languages.

CHAPTER 2

BRIEF PRESENTATION OF A PREDICATE CALCULUS

In this chapter, I give a brief, but sufficient for our purpose, presentation of a predicate calculus. A predicate calculus is an axiomatized formal presentation of predicate logic. The reader who is familiar with the construction of predicate calculus may skip the presentation below. I do, however, refer it in the following chapters.

The presentation is a summary of Enderton's *Mathematical Introduction to logic* [2] with some influence of Mendelsons *Introduction to Mathematical Logic* [10]. That is, the following definitions and theorems are from Enderton; definitions and theorems that are from Mendelson I will explicitly cite.

A formal presentation of predicate calculus has three primary sections: a syntax, a semantics, and an inference system. The syntax of predicate calculus lists the symbols that the language of predicate calculus contains and provides formation rules that generate grammatically well formed expressions. The semantics of predicate calculus provides meaning for the symbols of the language and a theory of truth and validity. The syntax and the semantics comprise the formal language of the logic. The inference system provides a set of axioms and rules of inferences for the language.

2.1 THE SYNTAX

The syntax of predicate calculus begins with what the language contains, the symbols of the language. The symbols of the language are divided into parameters and logical symbols. The parameter symbols are void of meaning and require a formal semantics for meaning. On the contrary, the logical symbols have a fixed meaning. The symbols of the language are given below.

Definition 2.1. *Let $\mathcal{L}$ be a language that has the following:*

Parameters

1. *A set of individual constant symbols:* $a_1, a_2, ..., a_k, ...$
2. *A set of n-place predicate symbols:* $P_1, P_2, ..., P_k, ...$
3. *A set of n-place function symbols:* $f_1, f_2, ..., f_k, ...$
4. *Quantifier symbol:* $\forall$.
   Logical symbols
5. *Punctuation marks: the left parenthesis '(', the right parenthesis ')', and the comma ','.*

6. *Propositional connective symbols: The negation symbol $\sim$, which means 'not', the implication connective $\rightarrow$, which means 'implies'.*
7. *A set of individual variables: $x_1, x_2, ..., x_k, ...$*

Note that in our language we have individual variables and individual constants. Both are considered terms in our language. A term in predicate calculus is a noun that represents an individual. The formal definition of term is given bellow.

Definition 2.2. *Terms in $\mathcal{L}$ are defined as follows:*

1. *any individual constant a is a term*
2. *any individual variable x is a term*
3. *if each t_i is a term and f is an n-place function symbol, then the expression $f(t_1, ..., t_n)$ is a term.*

An *expression* is any finite sequence of symbols. So a term is an expression. Another kind of expression is a formula. A formula is an expression generated by a certain set of rules. These rules are the *formation rules* of $\mathcal{L}$. The most basic formulas of $\mathcal{L}$ are called the atomic formulas, and they are generated by the following formation rule. If P is a predicate symbol and $a_1, a_2, ..., a_n$ are individual symbols (symbols that represent individuals) then $P(a_1)$, $P(a_1, a_2)$, $P(a_1, a_2, ..., a_n)$ are atomic formulas of $\mathcal{L}$. From the atomic formulas we can then build compound and quantified formulas. The formation rules are given below.

Definition 2.3. ***Formation Rules*** *The set of formulas of $\mathcal{L}$ is defined by the following rules:*

1. *Atomic formula: If P is an n-ary predicate symbol and $t_1, ..., t_n$ are terms, then $P(t_1, ..., t_n)$ is a formula of $\mathcal{L}$.*
2. *Negation. If ϕ is a formula, then $\sim \phi$ is a formula.*
3. *Condition. If ϕ and ψ are formulas, then $\phi \rightarrow \psi$ is a formula.*
4. *Quantification. If ϕ is a formula and x is a variable, then $\forall x \phi$ is a formula.*

Only expressions which can be obtained by finitely many applications of rules (1)-(4) are the formulas of $\mathcal{L}$. In other words, the formulas of $\mathcal{L}$ are those expressions that can be built up from the atomic formulas by applications (zero or more times) of the connective and the quantifier symbol.

A variable x may be bound in the scope of a quantifier as in the formula $\forall x P(x)$, or x may be free from the scope of any quantifier as in the formulas $P(x)$ and $\forall y P(x, y)$. The definition of a free versus a bound variable is given below.

Definition 2.4. ***Free versus Bound Variable*** *An occurrence of x is bound in a formula ϕ iff either it is the occurrence of x in a quantifier $\forall x$ in ϕ or it lies within the scope of the quantifier $\forall x$ in ϕ. Otherwise, the occurrence of x is free in ϕ.*

The following definition is from Mendelson.

Definition 2.5. *If ϕ is a formula and t is a term, then t is said to be free for x in ϕ if no free occurrence of x_i in ϕ lies within the scope of any quantifier $(\forall x_j$, where x_j is a variable in t. In particular, t is free for x in a formula $\phi(x)$ means that: if t is substituted for all free occurrences (if any) of x in $\phi(x)$, then no occurrence of a variable in t becomes a bounded occurrence in $\phi(t)$.*

Definition 2.6. ***Sentence*** *If no variable occurs free in the formula ϕ then ϕ is a* sentence. *We usually let σ denote a sentence.*

This completes the syntax portion of $\mathcal{L}$. We now want meaning for the symbols of $\mathcal{L}$ and a theory of truth for $\mathcal{L}$. This is treated in the semantics of $\mathcal{L}$.

2.2 The Semantics

The syntax of a formal language contains symbols and rules of formation. The symbols are void of meaning and therefore require a semantics. Language $\mathcal{L}$ is considered the object language, since it is the language we are interested in observing and exploring. Their representations, which are treated in the semantic portion of $\mathcal{L}$, are given in a meta-language, and this meta-language is usually some natural language. Specifically, the language of the semantics of $\mathcal{L}$ is English.

The semantics of $\mathcal{L}$ assigns meaning to the symbols of $\mathcal{L}$ by providing an interpretation that translates like a foreign language dictionary from one language to another language all together. The interpretation is a mapping from the formal language of $\mathcal{L}$ to some natural language like English. In particular it maps the parameters individual symbols and predicate symbols of $\mathcal{L}$ to the words or phrases of a natural language that respectively stand for individuals and predicate phrases in that natural language. The semantics of $\mathcal{L}$ is given formally as follows.

Definition 2.7. ***Interpretation and the Universe****. Let $\mathfrak{A}$ be an* interpretation *and let the nonempty set **D** be the* universe *(*domain of discourse*). The interpretation $\mathfrak{A}$ is a function whose domain is the set of parameters $\mathcal{L}$ such that:*

1. *Each constant symbol a is assigned to an individual $\mathfrak{A}(a)$ in **D**, which $\mathfrak{A}(a)$ is an individual entity in **D**;*
2. *Each predicate symbol P^n of arity n is assigned a relation $\mathfrak{A}(P^n)$ over $\mathbf{D}^n$ (in particular $\mathfrak{A}(P) \subseteq \mathbf{D}$);*
3. *Each function symbol f^n of arity n is assigned a function $\mathfrak{A}(f^n)$ from $\mathbf{D}^n$ to **D**.*

A brief example will illustrate our definition above. Let the universe **D**$= \{2, 3\}$, $\mathfrak{A}(a_1) = 2$, $\mathfrak{A}(a_2) = 3$, $\mathfrak{A}(P_2) =$'$>$' (the greater than sign), and $\mathfrak{A}(f_2) =$'+' (the addition sign). Then $P_2(a_2, a_1)$, or $a_2 P_2 a_1$, and $f_2(a_1, a_2)$, or $a_1 f_2 a_2$, are formulas where $\mathfrak{A}(a_2)\mathfrak{A}(P_2)\mathfrak{A}(a_1)$ means $3 > 2$ and $\mathfrak{A}(a_1)\mathfrak{A}(f_2)\mathfrak{A}(a_2)$ means $3 + 2$.

Now we seek a concept of truth. We want to establish what is true in an interpretation $\mathfrak{A}$. In order to develop the concept of truth, we first define a more general concept, and that is the concept of satisfaction. From the idea of satisfaction we build the idea of truth. More precisely, we want to define what it means for $\mathfrak{A}$ to *satisfy* ϕ with s, in symbols $\models_{\mathfrak{A}} \phi[s]$.

We begin by giving meanings to formulas with free variables such as $P(x)$ with an object assignment which assigns the variable of the formula to an element in the universe.

Definition 2.8. ***Object Assignment*** *An object assignment s is a function $s : \mathbb{X} \rightarrow$ **D** from the set $\mathbb{X}$ into **D**, where $\mathbb{X}$ is the set of all variables.*

Now given an object assignment s, we extend the function s to the function $\bar{s}$, where $\bar{s} : \mathbb{T} \rightarrow$ **D** is a function from the set $\mathbb{T}$ of all terms into the universe **D**, noting that $\mathbb{T}$ is a superset of $\mathbb{X}$. The function $\bar{s}$ is defined as follows.

Definition 2.9. ***Extension of Object Assignment***

1. *For each individual variable x, $\bar{s}(x) = s(x)$;*
2. *For each individual constant a, $\bar{s}(a) = \mathfrak{A}(a)$;*
3. *If $t_1, ..., t_n$ are terms and f is an n-place function symbol, then $\bar{s}(f(t_1, ..., t_n)) = \mathfrak{A}(f)(\bar{s}(t_1), ..., \bar{s}(t_n))$*

Definition 2.10. ***Satisfaction****. Let $\mathfrak{A}$ be a an interpretation for $\mathcal{L}$ and s an object assignment. We shall define $\models_{\mathfrak{A}} \phi[s]$ to mean for $\mathfrak{A}$ to satisfy ϕ with s as follows:*

Atomic Formulas

1. *For an n-place predicate parameter P, $\models_{\mathfrak{A}} P(t_1, ..., t_n)$ iff $(\bar{s}(t_1), ..., \bar{s}(t_n)) \in \mathfrak{A}(P)$;*
   Other Formulas
2. *$\models_{\mathfrak{A}} \sim \phi[s]$ iff $\not\models_{\mathfrak{A}} \phi[s]$;*
3. *$\models_{\mathfrak{A}} (\phi \rightarrow \psi)[s]$ iff either $\not\models_{\mathfrak{A}} \phi[s]$ or $\models_{\mathfrak{A}} \psi[s]$ (or both);*
4. *$\models_{\mathfrak{A}} \forall x \phi[s]$ iff for every $d \in$ **D**, we have $\models_{\mathfrak{A}} \phi[s(x|d)]$.*

The function $s(x|d)$ is exactly like the function s with one exception: At the variable x, the function $s(x|d)$ assumes the value d. This is expressed by the following equation

$$s(x|d)(y) = \begin{cases} s(y) & if\ y \neq x \\ d & if\ y = x \end{cases}$$

The function $s(x|d)$ is the same as $s(x)$ except the variable x assumes the value d in **D**.

Notice that $\forall$ means more specifically for all things in **D**.

Now we verify that the notion of satisfaction depends on the values d in **D** not on the function s, i.e. any information on the function s ought not to matter at all. This is captured in the following theorem.

Theorem 2.1. *Suppose s_1 and s_2 are functions from $\mathbb{X}$ into **D** which agree at all variables that occur free in the formula ϕ. Then*

$\models_{\mathfrak{A}} \phi[s_1]$ *iff* $\not\models_{\mathfrak{A}} \phi[s_2]$

(For proof see p.86 Enderton)

There is an analogous fact for interpretations: If $\mathfrak{A}$ and $\mathfrak{B}$ agree at all the parameters that occur in ϕ, then $\models_{\mathfrak{A}} \phi[s]$ iff $\models_{\mathfrak{B}} \phi[s]$.

The corollary below gives the key result of satisfaction of sentences of which we base the definition of truth.

Corollary 2.1. ***A**. For a sentence σ, either*

1. *$\mathfrak{A}$ satisfies σ with every function s, or*
2. *$\mathfrak{A}$ does not satisfy σ any such function s.*

Definitions. If (1) holds, then we say that the sentence σ is *true* in $\mathfrak{A}$, and this written as $\models_{\mathfrak{A}} \sigma$. We also say that $\mathfrak{A}$ is a *model* of σ. If (2) holds, then we say that σ is *false* in $\mathfrak{A}$, written as $\not\models_{\mathfrak{A}} \sigma$. (Note that (1) and (2) cannot both hold since **D** is nonempty.) Let Σ be a set of sentences, then $\mathfrak{A}$ is a *model* of Σ iff it is a model of every member of Σ.

We are now prepared to formulate important concepts in logic.

Definition 2.11. ***Logical Implication**. Let Γ be a set of formulas and let ϕ be a formula. Then,* Γ logically implies ϕ, *written $\Gamma \models \phi$, iff for every structure $\mathfrak{A}$ for the language and every function $s : \mathbb{X} \to$ **D** where $\mathfrak{A}$ satisfies every member of Γ with s, $\mathfrak{A}$ also satisfies ϕ with s.*

Definition 2.12. ***Logical Equivalence**. Let ϕ and ψ be a formulas. Then, ϕ and ψ are* logically equivalent *iff $\phi \models \psi$ and $\psi \models \phi$.*

Definition 2.13. ***Validity of Formulas**. Let ϕ be a formula. Then, ϕ is* valid *iff for every $\mathfrak{A}$ and every $s : \mathbb{X} \to$ **D**, $\mathfrak{A}$ satisfies ϕ with s. (Thus, ϕ is valid iff $\emptyset \models \phi$, written simply as $\models \phi$.)*

Corollary 2.2. ***B**. **Validity of Sentences** Let Σ be a set of sentences and σ a sentence. Then, $\Sigma \models \sigma$ iff every model of Σ is also a model of σ. The sentence σ is* valid *iff it is true in every interpretation.*

This completes the semantic portion of our language and, hence, completes the formal language of a predicate calculus. We will now provide an inference system for our language.

2.3 THE INFERENCE SYSTEM

An inference system (aka a deductive system) consists of axioms and rules of inference that are used to derive theorems and comprise a formal theory. A formal theory in predicate calculus is the set of axioms and all the theorems derived from the axioms. In other words, following Enderton, for any interpretation $\mathfrak{A}$, a theory of $\mathfrak{A}$, written Th$\mathfrak{A}$, is the set of sentences true in $\mathfrak{A}$. I will use Mendelson's notation and let K be a formal theory, a first-order theory, of $\mathfrak{A}$ (so K = Th$\mathfrak{A}$) in the language of $\mathcal{L}$. A formal theory K whose symbols and formulas are the symbols and formulas of $\mathcal{L}$ contains a set of axioms, which are formulas of $\mathcal{L}$. There are two kinds of axioms: the logical axioms and the proper (non-logical) axioms. Proper axioms vary from theory to theory; the axioms of group theory and the axioms of number theory are proper axioms. The logical axioms are axioms for the inference system itself. The logical axioms of K are given below, which are taken from Mendelson.

A first-order theory in which there are no proper axioms is called a first-order predicate calculus.

Definition 2.14. ***Logical Axioms*** *Let ϕ and ψ be formulas and let x be a variable. The set Λ of logical axioms consists of the following:*

A1 $\phi \to (\psi \to \phi)$

A2 $(\phi \to (\psi \to \chi)) \to ((\phi \to \psi) \to (\phi \to \chi))$

A3 $(\sim \psi \to\sim \phi) \to ((\sim \psi \to \phi) \to \psi)$

A4 $\forall x\phi(x) \to \phi(t)$*, where* t *is free for* x *in* $\phi(x)$.

A5 $\forall x(\phi \to \psi) \to (\phi \to \forall x\psi)$*, where* ϕ *contains no free occurrences of* x.

A6 $\forall x(x = x)$

A7 $x = y \to (\phi(x, x) \to \phi(x, y)$*, where* $\phi(x, y)$ *arises from* $\phi(x, x)$ *by replacing some, but not necessarily all, free occurrences of* x *by* y*, provided that* y *is free for* x *in* $\phi(x, x)$.

Axioms (A1) - (A5) are the logical axioms. The last two axioms are axioms of equality which may or may not be included in Λ. These axioms are included just in case the language has a predicate which denotes equality.

Let ϕ and ψ be formulas, then ϕ is a *generalization* of ψ if and only if for some $n > 0$ and some variables $x_!, ..., x_n$, $\phi = \forall x_1, ..., \forall x_n\psi$. The case when $n = 0$ allows $\phi = \psi$ which means that any formula is a generalization of itself. The logical axioms are thus all generalizations of the formulas A1-A7.

The formal theory K will have two rules of inference and that is modus ponens and Generalization.

Rule of Inference:

1. Modus Ponens. ψ follows from $\phi \to \psi$ and ϕ. The abbreviation MP is used to indicate application of this rule.
2. Generalization: $\forall x\phi$ follows from ϕ. The abbreviation UG is used to indicate application of this rule.

Now we are ready to define a deduction in our inference system and consequently we make precise the definition of proof and theorem.

Definition 2.15. *A* deduction *of a formula ϕ from a given set of formulas Γ (possibly empty) is a sequence of formulas $\phi_1, \phi_2, ..., \phi_n$, where for each ϕ_i either $\phi_i \in \Lambda$ (ϕ_i is a logical axiom), $\phi_i \in \Gamma$ (ϕ_i is an assumption), or is obtained by some rule of inference on some of the preceding formulas in the sequence. Such a sequence is called a proof or a deduction of ϕ from Γ. The members of Γ are called the hypotheses or premisses of the proof. We use $\Gamma \vdash \phi$ to abbreviate 'ϕ is a deduction from Γ' or 'ϕ is a theorem of Γ. A theorem of Γ is a formula ϕ of Γ such that ϕ is the last formula of some proof or deduction in Γ. Thus, theorems of Γ are formulas which are obtained from $\Gamma \cup \Lambda$ by use of modus ponens (some rule of inference in general).*

Theorem 2.2. *Every formula ϕ of K that is an instance of a tautology is a theorem of K, and it may be proved using only axioms (A1)-(A3) and MP.*

The abbreviation Taut for 'Tautology' will indicate an application of Theorem 2.2 in a proof.

Now in order to increase our deductive arsenal so we may apply different tactics and strategies to prove theorems, we derive what we call meta-theorems, theorems of the deductive system. I give these theorems below without proof.

Theorem 2.3. ***Deduction Theorem (DT)*** *Let Γ be the set of formulas and Λ be the set of logical axioms. If $\Gamma; \phi \vdash \psi$, then $\Gamma \vdash (\phi \rightarrow \psi)$ [2].*

Other meta theorems are below, beginning with Enderton's 'Generalization Theorem', which I call 'Universal Generalization'.

The meta theorem below due to Mendelson's 'Particularization rule A4'. I call the meta theorem below 'Universal Specification'.

Theorem 2.4. *Universal Specification (US). If t is free for x in $\phi(x)$, then $\forall x \phi(x) \vdash \phi(t)$.*

The abbreviation US is used to indicate application of this rule.

The meta theorem below due to Mendelson's 'Existential Rule E4'. I call the meta theorem below 'Existential Generalization'.

Theorem 2.5. *Existential Generalization (EG). Let t be a term that is free for x in a formula $\phi(x, t)$, and let $\phi(t, t)$ arise from $\phi(x, t)$ by replacing all fee occurrences of x by t. ($\phi(x, t)$ may or may not contain occurrences of t.) Then $\phi(t, t) \vdash \exists x \phi(x, t)$.*

The meta theorem below is taken directly from Mendelson's 'Rule C', which I call 'Existential Specification'. Rule C allows us to deduce from $\exists x \phi(x)$ to $\phi(a)$ for an arbitray constant a. The abbreviation EG is used to indicate application of this rule. For Rule C I refer to Mendelson p. 82 [10]. Below is an abridged version.

Theorem 2.6. *Existential Specification (ES). A rule C deduction in K:* $\Gamma \vdash_C \phi$ *if and only if there is a sequence of formulas* $\chi_1, ..., \chi_n$ *such that* χ_n *is* ϕ *and there is a preceding formula* $\exists x \psi(x)$ *such that* χ_i *is* $\psi(a)$*, where* a *is a new individual constant.*

Our inference system is now equipped with enough rules of deductions so that we may choose in accordance to some strategy to prove $\Gamma \vdash \phi$.

The inference system of a first order predicate calculus has the properties of being consistent, sound, and complete. These definitions are given below.

Consistency: A set Γ of formulas is consistent if and only if there is no formula ϕ such that ϕ and $\sim \phi$ are elements of Γ. Otherwise Γ is inconsistent, i.e. both ϕ and $\sim \phi$ are theorems in Γ.

Soundness Theorem states that deductions lead to valid or true conclusions. In other words deductions are truth preserving. In symbols If $\Gamma \vdash \phi$ then $\Gamma \models \phi$.

Completeness Theorem is the converse of the soundness theorem. If $\Gamma \models \phi$ then $\Gamma \vdash \phi$.

Compactness In particular, a set of sentences has a model iff every finite subset has a model.

This completes our brief presentation of a first order predicate calculus.

CHAPTER 3

THE GRAMMAR OF AN ATOMIC FORMULA AND THE SYNTAX OF A RELATION

By grammar of an atomic formula, I mean specifically its syntactic and semantic components. Semantically, atomic formulas generally represent relations. That is, for any interpretation, the general semantic denotation of the atomic formula in predicate calculus is a relation, specifically a relation on individuals. By relation I mean primarily in the mathematical sense, which is an ordered n-tuple (where n is a natural number) and a subset of a cartesian product; the number n corresponds to the arity of the relation which is the number of objects being related. For example, if we let R be a relation with arity 2 (2-tuple or 2-place) and let e_1 and e_2 be individual entities, then, a 2-tuple, or binary, relation may be mathematically written as $R(e_1, e_2)$ or e_1Re_2 which is read as 'e_1 relates to e_2'.

Syntactically, the atomic formula is made up of a predicate and terms. The syntactic part, predicate, corresponds to the semantic part(s), relation, and the syntactic part(s), term(s), corresponds to the semantic part, individual(s). For example, consider the atomic formula in predicate calculus $P(x, y)$ where P is a predicate and x and y are terms. Semantically, by any interpretation, the predicate P generally represents a relation, and the terms x and y generally represent individuals. The atomic formula $P(x, y)$ thus represents a relation of two individuals. The atomic formula may be viewed as characterizing the syntax of a relation on individuals as a predicate on terms. Thus, we may say that the predicate is the syntax of a relation.

I do not abandon the intuitive (as opposed to mathematical) sense of the word relation found in natural languages. Intuitively, a relation is a way in which two or more things, or ideas, are connected. I assume that the general semantic representation of a simple statement in a natural language is a relation as well. When we explore examples of simple statements in a natural language (English), we will find that the predicate on terms syntactic structure or characterization falls short in characterizing the relations of the simple statements. This is because a predicate on terms characterization is limited to only a certain kind of relation, a relation involving only individuals. It, furthermore, is at odds with the intuitive sense of a relation when we consider examples of simple statements in a natural language that are modeled by the monadic predicate.

My purpose in this chapter is to show that the verb on nouns syntactic structure for atomic formulas is a more natural syntactic structure than the predicate on terms. We will explore examples of simple sentences in English (representable with atomic formulas) that illustrate a conflict with the intuitive meaning of a relation and how predicate calculus characterizes the syntax of a relation (i.e. the syntactic structure of the atomic formula), which is a predicate on terms syntactic structure. These examples will, furthermore, reveal an alternative way, a more natural way, to characterize the syntax of a relation, which is a verb on nouns syntactic structure. This characterization is not limited to relations on individuals and therefore can model the relations of simple statements where a predicate on terms characterization fails. It moreover accords with the intuitive and natural sense of what we mean by relation. This characterization thus provides an alternative way to constitute the syntax of atomic formula, which is more natural and, consequently, gives way to a deeper syntax of a relation.

Consider the monadic predicate atomic formula, Ps, in predicate calculus, where s is a term (individual term) and P is a monadic predicate, a 1-place predicate. Monadic predicates represent what is called unary (1-place) relations. In predicate calculus, a unary relation is a special relation. Semantically, the notion of property is defined as a unary relation. A property is some attribute to an individual; specifically a property can be a class, quality, or quantity. The monadic sentence, Ps, represents in a natural language a simple grammatical sentence with an intransitive verb. The syntactic parts of Ps is the predicate P and the term s of Ps; the term s, in this case, is also be considered the subject of the atomic formula Ps. Semantically, s denotes an individual entity and P denotes a property. The grammatical aspects of the monadic sentence are expressed in the table below:

Table 3.1. The Grammar of a Monadic Predicate Formula

	s	P
syntax	term	predicate
semantics	individual	property

Now let the predicate P in Ps contain the copula verb along with either a noun or adjective having the syntactic structures $P = V$ - N, or $P = V$ - A where V is the copula verb and N is a noun and A is an adjective (determiners ’a’ and ’the’ may be implied in the structures). The copula verb is the most important verb in logic. It is a kind of intransitive verb or linking verb; in English, the copula verb is any form of ’be’. If we consider simple statements with the copula verb where the subject is limited to individual entities (as opposed to properties), then we have three distant meanings for the copula verb. They are identity, class membership, and quality attribution. Below is a table of the three meanings of the copula each paired with a simple grammatical sentence in English as an example.

Table 3.2. Kinds of Meanings of the Copula with Examples

Copula Meaning	Sentence example
Identity	Clark Kent is superman.
Membership	Socrates is a man.
Attribution	Socrates is wise.

We will explore the kinds of meanings above through their respective simple sentences that exemplify them and compare the two syntactic characterizations, a verb on nouns versus a predicate on terms.

The first of these meanings is identity, which is exemplified by the simple sentence 'Bruce Wayne is Batman' or, the classical example, 'Hesperus is Phosphorus'. Identity in predicate calculus is not syntactically represented by the monadic predicate Ps. Rather, its represented by the dyadic predicate. The dyadic predicate is a two place predicate in predicate calculus, say, sPo or $P(s, o)$, where s is the first coordinate and o is the second coordinate of P, and P is the dyadic predicate and s and o are terms. In the example 'Clark Kent is superman', we may let s to represent 'Clark Kent', which in this case is the subject of the predicate, o to represent 'superman', which is the object of the predicate, and P to represent 'is', which is the predicate. Yet, as we can see, P is the verb of the sentence. Since the predicate P is identified as the verb V, we may simply interchange V for P. So, alternatively and more precisely, we can represent the sentence 'Clark Kent is superman' a relation with the syntactic structure sVo or $V(s, o)$, where the verb V is a relation of the terms (nouns) s and o, or i.e. V is a relation of subject s and object o. Thus, the identity statement can be characterized as a verb as a relation of two nouns (in symbols N_1-V-N_2 or $V(N_1, N_2)$) which is a more precise way to characterize the syntax of a relation than a predicate on terms.

Unlike the similarity of the syntactic characterizations above, when we explore the other two meanings of the copula, we find different syntactic structures all together when viewing a verb as a relation versus a predicate as a relation. This is because, the sentences that denote class membership and quality attribution have monadic predicate representations in predicate calculus, such as Ps. Monadic predicates represent unary relations, and the very idea of a unary relation is at odds with the intuitive sense of relation, since the intuitive definition requires a relation to relate at least two things. Yet, a unary relation being a property relation naturally implies a relation of two things, specifically a relation of property and individual, which is naturally a binary (2-place) relation in structure. When we view the verb as a relation for the same sentences we will see that the relation is a binary relation of individual entity and property.

Consider the sentence 'Socrates is a man' which is a simple sentence with the copula denoting membership. We may let s be 'Socrates' and let P be the predicate 'is a man' then we have the monadic predicate formula Ps that represents a sentence denoting class membership. The monadic predicate not only conflicts with the intuitive sense of relation, but it also conflicts with the mathematical idea of membership as binary relation. Now, if we let s be 'Socrates' but let M be the noun 'man' (or noun phrase 'a man') and let V be the verb 'is' then we have $V(s, M)$, or sVM, where the verb is a relation on nouns (or noun phrases, in general) and a noun may either be an individual or property. This accords with what we intuit when observing the sentence 'Socrates is a man': a binary relation between two nouns 'Socrates' and 'man', specifically a class membership relation. Thus, in this case, the verb on nouns syntactic structure has a one-to-one correspondence with the semantics of the sentence, the binary relation of individual and property, whereas the predicate on terms does not.

The sentence which denotes attribution has a similar conclusion. The sentence 'Socrates is wise' has the monadic predicate representation Ps in PC when we let s be 'Socrates' and P be the predicate 'is wise'. Yet we may view the binary relation naturally inherent in the sentence by letting s be 'Socrates', V be the verb 'is', and W be the adjective 'wise' and thus have the syntactic structure of the relation as $V(s, W)$ or s-V-W where the verb is a relation of noun and adjective. (In general I say that the verb is a relation on nouns; for simplicity, I include adjectives as nouns.) This view point, the verb as a relation, clearly reveals once again the semantics of the sentence as a binary relation of individual entity and property. Hence, this is a case where the monadic predicate falls short in modeling the binary relation inherent in a simple sentence, and, hence, this is a case where a verb on nouns syntactic structure successfully models the binary relation and thus accords well with natural syntax with a one to one correspondence.

Note that as a consequence of viewing the verb as the syntax of a relation the notion of property is, syntactically, a noun (or noun phrase), and this notion differs from predicate calculus notion of property. In predicate calculus, the notion of property is associated with the predicate of a monadic atomic formula. These different characterizations of property reflects the competing logical viewpoints of what a property is or how a property relation is characterized syntactically [12]. Some like Frege prefer the notion of property as a predicate, which is taken as fundamental in predicate calculus. Other logicians contend that property, syntactically, is only the noun or noun phrase of the predicate; the copula verb is not included.

If we explore simple sentences where the verb is transitive, we will arrive at a conclusion that is similar to the one in the first example, the simple sentence where the copula denotes the identity relation. That is, the view of the predicate as a relation is virtually identical to the view of the verb as a relation, since the predicate is the verb. Consider the sentence ’Bogart kisses Bacall’. In predicate calculus this may be represented in symbols as $K(b_1, b_2)$ where b_1 stands for ’Bogart’, b_2 stands for ’Bacall’, and K is the predicate that stands for ’kisses’. By simply shifting our view point to K as the verb that stands for ’kisses’, we see that the predicate as a relation is the same as the verb as relation, a relation of nouns and a relation of subject and object, $V(S, O)$. This is because, in predicate calculus the predicate is considered the verb in a simple transitive sentence.

The sentence ’John gives Mary a flower’ is also represented in predicate calculus where the predicate is the verb of the sentence. Let j denote ’John’, f denote ’flower’, m denote ’Mary’, and let G be the predicate that denotes ’gives’, then we have $G(j, f, m)$. By viewing the verb as a relation we have the verb as a trinary relation of nouns and a relation of subject, direct object, and indirect object, $V(S, O_d, O_i)$.

The table below shows the two syntactic characterizations of a relation for the following sentences:

Table 3.3. Predicate as a Relation versus Verb as a Relation

Sentence	Predicate Atomic Formula	Verb Atomic Formula
Clark Kent is superman.	is(Clark Kent, superman)	is(Clark Kent, superman)
Bogart kisses Bacall.	kisses(Bogart, Bacall)	kisses(Bogart, Bacall)
John gave Mary a flower.	gave(John, a flower, Mary)	gave(John, a flower, Mary)
Socrates is a man.	is a man(Socrates)	is (Socrates, a man)
Hillary is the mother of Chelsea.	is the mother of(Hilary, Chelsea)	is (Hilary, the mother of, Chelsea)

If we continue to investigate the kinds of simple statements in a natural language such as English, I believe we can maintain the view that, syntactically, the verb is a relation on nouns.

CHAPTER 4

The General Formal Language

My purpose is to construct a formal language based on the idea that the syntax of a relation is a verb on nouns, where nouns denote individuals or properties. So, verbs and nouns are the basic constituents of the atomic formulas in this formal language, where the verb on nouns is the syntactic structure of the atomic formulas.

The verb as a relation has arity, which is the number of arguments a relation takes. Arity of the verb will correspond to what is called the valency of the verb. By valency, I mean specifically in the linguistic sense when applied to a verb. The valency of the verb is the least number of arguments required to make a grammatical sentence. Herein, the valency of the verb determines the number of arguments the verb takes as a relation of nouns or noun phrases. For example, the valency of a transitive verb is 2, since the verb requires a direct object as well as a subject; the valency of an intransitive verb, such as 'walk', is 1. (Valency in the linguistic sense is further discussed in Chapter 6.)

The formal language is established as a first-order theory. This is so we may construct a first-order predicate calculus from the formal language. The construction of the formal language is divided into the usual three parts: the syntax, the semantics, and inference system.

4.1 The Syntax

Let the symbol $\mathcal{L}_{\mathcal{V}}$ be the formal language where the verb on nouns is the syntactic structure of the atomic formula.

Definition 4.1. *Let $\mathcal{L}_{\mathcal{V}}$ be a language that has the following:*

Parameters

1. *A set of noun symbols: $N_l^{r_1}, N_2^{r_2} ..., N_k^{r_k}, ...$ where $r_1, r_2, ..., r_k, ...$ are the ranks (defined below) of the noun symbols.*
2. *A set of n-ary verb symbols: $V_1, V_2, ..., V_k, ...$ where n is the arity of the verb symbol.*
3. *A set of function symbols: $f_1, f_2, ..., f_k, ...$*
4. *Quantifier symbol: $\forall$.*

Logical symbols

5. *Punctuation marks: the left parenthesis '(', the right parenthesis ')', and the comma ','.*

6. *Propositional connective symbols: The negation symbol* $\sim$, *the implication symbol* $\rightarrow$.
7. *a set of individual variables:* $x_1, x_2, ...$

The notion of rank for a noun is used to distinguish between a concrete noun and an abstract noun. Herein a noun is either concrete or abstract. The idea of rank is given explicitly below.

Definition 4.2. ***Rank of a Noun***
Let N^r *be a noun symbol, then the* rank *of the noun,* r*, where* $r = 0, 1$*, indicates whether the noun is a* concrete noun *or* abstract noun*: if* $r = 0$*, then* N^r *is a concrete noun; and if* $r = 1$*, then* N^r *is an abstract noun.*

We will see later, in the semantics of our language, that a concrete noun represents an individual entity and an abstract noun represents a property.

A term in predicate calculus is a noun which is limited to denote only individuals. Terms in the language $\mathcal{L}_{\mathcal{V}}$ are nouns that are either abstract or concrete. Thus, terms in $\mathcal{L}_{\mathcal{V}}$ denote properties as well as individuals. In language $\mathcal{L}_{\mathcal{V}}$ *terms* are synonymous with *nouns*. Bellow is the formal definition of terms (nouns) in our language.

Definition 4.3. ***Terms (Nouns)***
The set of terms is defined as follows:

1. *Any noun symbol* N^r *is a term (noun). Specifically, any concrete noun symbol* N^0 *is a term (noun), and any abstract noun symbol* N^1 *is a term (noun).*
2. *Any individual variable* x *is a term (noun).*
3. *if each* $t_i^{r_i}$ *is a term (noun) and* f *is an* n*-place function symbol, then* $f(t_1^{r_1}, ..., t_n^{r_n})$ *is a term (noun).*

Only expressions which can be obtained by finitely many applications of rules 1, 2, and 3 are terms (nouns). For example, no expression involving a verb symbol is a term.

Remark: In language $\mathcal{L}_{\mathcal{V}}$ the words terms and nouns are used synonymously. If I want to emphasize some logical relation I use the word *term* over *noun*. If I want to emphasize some grammatical relation I use the word *noun* over *term*.

In the formal language of predicate calculus the atomic formula is made out of a predicate and terms. The n-ary predicate symbol P applied to terms letters $t_1, ..., t_n$ yield the n-ary formula $P(t_1, ..., t_n)$. In our language the atomic formula is made out of a verb and terms (nouns). The n-ary verb letter V applied to terms (nouns) $t_1, ..., t_n$ yield the n-ary formula $V(t_1, ..., t_n)$. This and other formulas of language $\mathcal{L}_V$ are expressed below.

Definition 4.4. ***Formation Rules***
The set of formulas of $\mathcal{L}_V$ *is defined as follows:*

1. *Atomic formula. If V is an n-ary verb symbol and $t_1, ..., t_n$ are terms, then $V(t_1, ..., t_n)$ is a formula of $\mathcal{L}_V$. In particular, $V(N_1^{r_1}, ..., N_n^{r_n})$ is a formula of $\mathcal{L}_V$, where $N_1^{r_1}, ..., N_n^{r_n}$ are noun symbols.*
2. *Negation. If ϕ is a formula, then $\sim \phi$ is a formula.*
3. *Implication. If ϕ and ψ are formulas, then $\phi \to \psi$ is a formula.*
4. *Quantification. If ϕ is a formula and x is a variable, then $\forall x\phi$ is a formula.*

Only expressions which can be obtained by finitely many applications of rules (1)-(4) are formulas of $\mathcal{L}_V$.

The atomic formulas are of special interest in our language. Below are important kinds of atomic formulas that we will encounter:

$V(N_l^r, N_m^s)$ is a formula, whenever $\mathcal{L}_V$ contains N_l^r and N_m^s and V with arity 2,

$V(N_l^r, N_m^s, N_n^t)$ is a formula, whenever $\mathcal{L}_V$ contains N_l^r, N_m^s and N_n^t and V with arity 3,

$V(x, N_m^s)$ is a formula, whenever $\mathcal{L}_V$ contains x and and V with arity 2,

$V(x, y)$ is a formula, whenever $\mathcal{L}_V$ contains x and y and V with arity 2.

Definition 4.5 (**Free versus Bound Variable**). *An occurrence of x is bound in a formula ϕ iff it lies within the scope of the quantifier $\forall x$ in ϕ. Otherwise, the occurrence of x is free in ϕ.*

For example, a variable x may be bound in the scope of a quantifier as in the formula $\forall xP(x)$, or x may be free from the scope of any quantifier as in the formulas $P(x)$ and $\forall yP(x, y)$. The definition of a free versus a bound variable is given below.

Definition 4.6 (**Sentence**). *If no variable occurs free in the formula ϕ then ϕ is a sentence. We let σ symbolize a sentence.*

In the language $\mathcal{L}_V$, the atomic formula has more structure than the atomic formula in predicate calculus. In particular, the language provides a deeper grammatical structure for the atomic formula. By taking the verb as a relation for the constitution of the atomic formula, we are in the position to establish grammatical relations in $\mathcal{L}_V$. The grammatical relations (aka grammatical functions) are the subject and the object (or objects) as well as the verb of a sentence. I define the grammatical relations subject, direct object, and indirect object according to the place coordinate of the verb as a relation.

Definition 4.7. ***Subject and Object***

The subject *(or* subject term*) of the verb is the noun (term) in the first coordinate of the verb; the* object *(or* object term*) or* direct object *of the verb is the noun (term) which occupies the second coordinate of the verb; the* indirect object *of the verb is the noun (term) which occupies the third coordinate of the verb.*

For example, if $V(N_l^r, N_m^s)$ is a formula of $\mathcal{L}_V$, then the *subject of* V is the first coordinate of V, N_l^r, and the *object of* V is the second coordinate of V, N_m^s; If $V(N_l^r, N_m^s, N_n^t)$ is a formula of $\mathcal{L}_V$, then the *subject of* V occupies the first coordinate of V, N_l^r, the *direct object of* V occupies the second coordinate of V, N_m^s, and the *indirect object of* V occupies the third coordinate of V, N_n^t.

Thus, the order underlying the verb as a relation defines the subject and object(s). Hence, we may say that the verb is a relation of subject and object(s).

To complete the syntax part of our language, we adopt the usual abbreviations for conjunction, disjunction and existential:

$\phi \wedge \psi$ for $\sim (\phi \rightarrow \sim \psi)$; $\phi \vee \psi$ for $\sim \phi \rightarrow \psi$; and $\exists x \phi$ for $\sim \forall x \sim \phi$.

4.2 THE SEMANTICS

Here we provide a formal semantics for our language $\mathcal{L}_V$.

Definition 4.8. ***Interpretation for TC***

Let $\mathfrak{G}$ be an interpretation of $\mathcal{L}_V$, then:

1. *$\mathfrak{G}$ assigns to the quantifier symbol $\forall$ a nonempty set* **N***, where* **N** *is the universe of $\mathfrak{G}$.*
2. *$\mathfrak{G}$ assigns to each noun letter N_l^r of $\mathcal{L}_V$, where $r = 0$, to an individual $\mathfrak{G}(N_l^r)$, which is an element in* **N***. Thus,* **N** *is the set of individuals.*
3. *$\mathfrak{G}$ assigns to each noun letter N_l^r where r = 1 of $\mathcal{L}_V$ to a property $\mathfrak{G}(N_l^r)$, which is a subset of* **N***, or i.e. an element in $\mathscr{N}$ where $\mathscr{N}$ is a family of subsets of* **N***. Thus, $\mathscr{N}$ is the set of properties.*
   We let Φ^r be a symbol such that $\Phi^r =$ **N** *if $r = 0$ and $\Phi^r = \mathscr{N}$ if r = 1. Then we have in general $\mathfrak{G}(N_l^r) \in \Phi^r$ where $\mathfrak{G}(N_l^r) \in$* **N** *if r = 0 and $\mathfrak{G}(N_l^r) \in \mathscr{N}$ if r = 1. This notation is convenient in establishing the interpretation for the verb.*
4. *$\mathfrak{G}$ assigns to each n-place verb letter V of $\mathcal{L}_V$, an n-ary relation $\mathfrak{G}(V) \subseteq \mathbf{N}^n$, $\mathfrak{G}(V) \subseteq \mathscr{N}^n$, or, in general, an n-ary relation $\mathfrak{G}(V)$ is a subset of any combination of an n-place cartesian product of* **N** *and $\mathscr{N}$.*
   The cartesian product of which $\mathfrak{G}(V)$ is a subset depends on the rank of the terms under the verb symbol V. This is expressed in the following rule:
   Consider the atomic formula $V(t_1^{r_1}, ..., t_n^{r_n})$. $\mathfrak{G}$ assigns to the n-place verb letter V of $\mathcal{L}_V$, an n-ary relation $\mathfrak{G}(V) \subseteq (\Phi_n^{r_n})^n$ ($(\Phi_n^{r_n})^n = \Phi_1^{r_1} \times ... \times \Phi_n^{r_n}$), where each $t_i^{r_i}$ in $t_1^{r_1}, ..., t_n^{r_n}$ corresponds to each $\Phi_i^{r_i}$ in $\Phi_1^{r_1} \times ... \times \Phi_n^{r_n}$. For $t_i^{r_i}$, if $r_i = 0$ then $\Phi_i^{r_i} =$ **N** *and if $r_i = 1$ then $\Phi_i^{r_i} = \mathscr{N}$.*
   For example:
   - *for $V(x_1, ..., x_n)$, $\mathfrak{G}$ assigns to the n-place verb symbol V an n-ary relation $\mathfrak{G}(V) \subseteq \mathbf{N}^n$;*
   - *for $V(N_1^{r_1}, ..., N_n^{r_n})$, $\mathfrak{G}$ assigns to the n-place verb letter V an n-ary relation $\mathfrak{G}(V) \subseteq (\Phi^{r_n})^n$, where $(\Phi^{r_n})^n = \Phi^{r_2} \times \Phi^{r_1} \times, ..., \Phi^{r_n}$;*

- *for $V(x, N^1)$, $\mathfrak{G}$ assigns to the 2-place verb letter V an binary relation* $\mathfrak{G}(V) \subseteq \mathbf{N} \times \mathscr{N}$.

For the concept of truth, we provide a more general notion, which is the notion of satisfaction. The notion of satisfaction begins with an object assignment. An object assignment that maps a variable to an individual in the universe, providing a denotation (or value) for the variable. We then generalize the object assignment map by increasing its domain to include all terms.

Definition 4.9. ***Object Assignment***

Let $\mathbb{X}$ be the set of all variables. An object assignment *s is a function $s : \mathbb{X} \rightarrow \mathbf{N}$ from the set $\mathbb{X}$ into **N**.*

Let $\mathbb{T}$ be the set of all terms. We now define the extension s^* which is a function from the set of all terms $\mathbb{T}$ into the universe **N** ($s^* : \mathbb{T} \rightarrow \mathbf{N}$), if the term is a variable or has rank 0; or s^* is a function into the family of subsets of the universe $\mathscr{N}$ ($s^* : \mathbb{T} \rightarrow \mathscr{N}$), if the term has rank 1. The extension of the object assignment is as follows.

Definition 4.10. ***Extension of Object Assignment***

Given an object assignment s, we shall extend the mapping s to the mapping s^*, which is defined recursively as follows:

1. *For each variable x, $s^*(x) = s(x)$;*
2. *For each noun symbol N^r, $s^*(N^r) = \mathfrak{G}(N^r)$;*
3. *If $t_1, ..., t_n$ are terms and f is an n-place function symbol, then* $s^*(f(t_1, ..., t_n)) = \mathfrak{G}(f)(s^*(t_1), ..., s^*(t_n))$.

Definition 4.11. ***Satisfaction***

Let $\mathfrak{G}$ be a an interpretation for $\mathcal{L}_V$ and s an object assignment. We shall define $\models_{\mathfrak{G}} \phi[s]$ to mean $\mathfrak{G}$ satisfies *ϕ* with *s as follows:*

Atomic Formulas

1. *If ϕ is an atomic formula $V(t_1, ..., t_n)$, then the assignment s* satisfies *ϕ if and only if $(s^*(t_1), ..., s^*(t_n)) \in \mathfrak{G}(V)$; that is, if the n-tuple is in the n-place relation $\mathfrak{G}(V)$; in symbols, $\models_{\mathfrak{G}} V(t_1, ..., t_n)[s]$ iff $(s^*(t_1), ..., s^*(t_n)) \in \mathfrak{G}(V)$.*
   Other Formulas
2. *$\models_{\mathfrak{G}} \sim \phi[s]$ iff $\not\models_{\mathfrak{G}} \phi[s]$;*
3. *$\models_{\mathfrak{G}} (\phi \rightarrow \psi)[s]$ iff either $\not\models_{\mathfrak{G}} \phi[s]$* or *$\models_{\mathfrak{G}} \psi[s]$ (or both);*
4. *$\models_{\mathfrak{G}} \forall x\phi[s]$ iff for every $d \in$ **N**, we have $\models_{\mathfrak{A}} \phi[s(x|d)]$.*

The function $s(x|d)$ is exactly like the function s with one exception: At the variable x, the function $s(x|d)$ assumes the value *d*.

Definition 4.12. ***Truth and Model***
A formula ϕ is true *for the interpretation $\mathfrak{G}$ (written $\models_{\mathfrak{G}} \psi$) iff $\mathfrak{G}$ satisfies ϕ with every function s. A formula ϕ is* false *for $\mathfrak{G}$ iff no function s satisfies ϕ. An interpretation $\mathfrak{G}$ is said to be a* model *for a set Σ of formulas of $\mathcal{L}_V$ iff every formula in Σ is true for $\mathfrak{G}$.*

Now, for our language $\mathcal{L}_V$, we may include the satisfaction theorem and its two corollaries from the predicate calculus in chapter 1. Also from the predicate calculus, we adopt the concepts of logical implication, logical equivalence, and validity.

We are now ready to establish an inference system for $\mathcal{L}_V$.

4.3 INFERENCE SYSTEM

My purpose for this section is to establish an inference system that is the same for a predicate calculus. For our first-order language $\mathcal{L}_V$, I establish the same inference system as the inference system of the predicate calculus in chapter 1. I begin by adopting the following axioms for the language $\mathcal{L}_V$.

Definition 4.13. ***Logical Axioms*** *Let ϕ and ψ be formulas and let x be a variable. The set Λ of logical axioms consists of the following:*

A1 $\phi \to (\psi \to \phi)$

A2 $(\phi \to (\psi \to \chi)) \to ((\phi \to \psi) \to (\phi \to \chi))$

A3 $(\sim \psi \to \sim \phi) \to ((\sim \psi \to \phi) \to \psi)$

A4 $\forall x \phi(x) \to \phi(t)$*, where t is free for x in $\phi(x)$.*

A5 $\forall x(\phi \to \psi) \to (\phi \to \forall x \psi)$*, where ϕ contains no free occurrences of x.*

A6 $\forall x(x = x)$

A7 $x = y \to (\phi(x, x) \to \phi(x, y)$*, where $\phi(x, y)$ arises from $\phi(x, x)$ by replacing some, but not necessarily all, free occurrences of x by y, provided that y is free for x in $\phi(x, x)$.*

Axioms (A1) - (A5) are the logical axioms. The last two axioms are axioms of equality which we will need for our theory of terms.

For the language $\mathcal{L}_V$ we include the inference modus ponenes: ψ follows from ϕ and $\phi \to \psi$.

For the language $\mathcal{L}_V$, I include in its inference system the inference system described in Chapter 1, such as the definitions of generalization and also the theorems, namely the deduction theorem (DT), universal generalization (UG), universal specification (US), existential generalization (EG), and existential specification (ES).

Let K_V be the theory of $\mathfrak{G}$. That is, K_V contains the logical axioms and all theorems mentioned above. Hence K_V is a first-order theory based on language $\mathcal{L}_V$ which contains only the logical axioms, i.e. K_V has no proper axioms.

4.4 The Predicate Atomic Formula as an Abbreviation

For our language $\mathcal{L}_V$, I will establish a method of converting atomic formulas in $\mathcal{L}_V$, atomic formulas with a verb on nouns syntactic structure, to atomic formulas with a predicate on terms syntactic structure. The method is a method of abbreviating from a verb on nouns syntactic structure to a predicate on terms. The method of abbreviation below will show that language $\mathcal{L}_V$ may include the language of predicate calculus.

First, I give a method of how we may establish the monadic predicate below. Then I generalize to any n-ary predicate atomic formula.

Predicate Abreviation

Consider the atomic formula with the form $V(t^0, N^1)$ where t^0 is an individual term (which is either an individual variable or a concrete noun), N^1 is an abstract noun (property term), and V is an intransitive verb. Then we may write $P(t^0)$ for $V(t^0, N^1)$ where P has the (grammatical) structure $P = V\text{-}N^1$. That is, predicate P includes a verb V and an abstract noun N^1 as its grammatical structure. So $P(t^0)$ is the monadic predicate atomic formula of predicate calculus. Thus, $P(t^0)$ is an abbreviation for $V(t^0, N^1)$.

We may generalize this method of abbreviation as follows:

Abbreviation to Atomic Predicate Formulas

Let the atomic formula $V(t_1, ..., t_n)$. If $t_1, ..., t_n$ are concrete (individual) terms $t_1, ..., t_n = t_1^0, ..., t_n^0$ then $V = P$. If $t_1, ..., t_n$ has both concrete terms and abstract terms and V is a verb. We take the concrete terms $t_{i_1}^0, ..., t_{i_k}^0$ from $t_1, ..., t_n$, with order preserved, and take the abstract terms $N_{i_{k+1}}^1, ..., N_{i_n}^1$ from $t_1, ..., t_n$. Then let the symbol P be our predicate, having the structure $P = V\text{-}N_{i_{k+1}}^1\text{-...-}N_{i_n}^1$. Thus, we write the predicate atomic formula $P(t_{i_1}^0, ..., t_{i_k}^0)$ for the verb atomic formula $V(t_1, ..., t_n)$.

The formal language $\mathcal{L}_V$ with a predicate abbreviations for its atomic formulas is virtually the same as the formal language of a predicate calculus. I let $\mathcal{L}_P$ be the formal language of a predicate calculus. Thus, $\mathcal{L}_V$ with predicate abbreviations for its atomic formulas determines a formal language $\mathcal{L}_P$. Thus, with language $\mathcal{L}_P$, we may simplify the semantics of $\mathcal{L}_V$ to the semantics of a predicate calculus. We then may construct a first order predicate calculus from $\mathcal{L}_P$, and call it theory K_P, where K_P is equipped with the same inference system as the predicate calculus in chapter 1.

4.5 Concluding Remarks

We, first, assumed that the basic syntactic structure of the atomic formula is a verb on nouns (terms) relation. We then established a formal language $\mathcal{L}_V$ based on our assumption. Consequently, the verb on nouns (terms) syntactic structure of the atomic formula expanded the idea of a term in a formal language. A term is a noun which may denote properties (abstract noun) as well as individuals (concrete noun). The verb on nouns syntactic structure provides more structure to the atomic formula. As we will explore later, it provides a deeper grammatical structure which accords to natural syntax.

The formal language $\mathcal{L}_V$ with a predicate abbreviations for its atomic formulas is virtually the same as the formal language of a predicate calculus. Let $\mathcal{L}_P$ be the formal language of a predicate calculus. Thus, $\mathcal{L}_V$ with predicate abbreviations for its atomic formulas determines a formal language $\mathcal{L}_P$. Thus, with language $\mathcal{L}_P$, we may simplify the semantics of $\mathcal{L}_V$ to the semantics of a predicate calculus. We then may construct a first order predicate calculus from $\mathcal{L}_P$, and call it theory K_P, where K_P is equipped with an inference system of a predicate calculus.

The language $\mathcal{L}_V$ can, furthermore, be used to construct a calculus that is altogether different than predicate calculus. The language $\mathcal{L}_V$, because of the expanded notion of a term, can be used to model the propositions of term logic. Thus the language of term logic can be established by a formal language. Moreover, we can provide inference rules that are unique to the language of term logic, inference rules based on the relation of terms. Hence, term logic can be established by a formal calculus. This will show that our language is capable of expressing two different logics which gives way to the construction of different formal calculi.

CHAPTER 5

TERM CALCULUS

My purpose for this chapter is to provide a formal calculus for term logic. I begin with a formal language for term logic with the formal language $\mathcal{L}_V$ as its basis and call it $\mathcal{L}_T$. I then provide a formal theory for $\mathcal{L}_T$. This theory is unique to term logic; it contains an inference system based on the relation of terms. I call this theory, theory T.

Theory T is a formal calculus that mathematically models term logic. Term logic assumes propositions that are made out of two terms connected by the copula; term logic derives one proposition (the conclusion) from two other propositions (the premises) by an inference rule called a syllogism, or in other words the form of the deduction of the premises to the conclusion is a syllogism. Theory T has two basic rules of inference; they are based on the relation of the terms in given propositions. With these two basic inferences along with common logical inferences such as contraposition, we can derive all 24 valid syllogisms as theorems of T. Theory T, as a formal calculus for term logic, is specifically what I call a term calculus (TC).

In this chapter, I also show that the formal language $\mathcal{L}_T$ can model the language of contemporary existing term logics and I show that the contemporary term logics can be modeled by a term calculus. Specifically, I provide the beginnings a formal theory for Lukasiewicz axiomatic method of Aristotle's term logic. I also provide an explicit translation of the language of TC, $\mathcal{L}_T$, and the language of Sommers term logic in his *Calculus of Terms*.

5.1 A FORMAL LANGUAGE FOR TERM LOGIC

Here, I provide a formal language for term logic which is based on the language of $\mathcal{L}_V$. I call this language $\mathcal{L}_T$. Term logic only allows propositions which are made up of two terms, connected by the copula verb. Thus, the formal language $\mathcal{L}_T$ will only contain terms and a single verb. I establish the syntax of formal language $\mathcal{L}_T$ in the three subsections Terms, The Copula Verb, and Propositions.

Terms

A term is a noun or noun phrase. Recall in $\mathcal{L}_V$ a noun either is a concrete noun, which represents an individual, or an abstract noun, which represents a property. Here I use the word term over noun and provide a notion of rank for a term. This is expressed in the definition below.

Definition 5.1. ***Singular Term and Abstract Term***
In language $\mathcal{L}_T$, E^n, F^m, and G^l are terms with ranks n, m, and l. For the term E^n the letter n is the rank of the term E^n, *where n = 0, 1. If n = 0, then $E^n = E^0$ is a* singular term, *a term that represents an individual. Thus singular terms have rank 0, and we shall abbreviate singular terms to lower case letters, for example we write e for E^0. Thus, the lower case letters e, f, g are singular terms. If the rank is $n = 1$, then $E^n = E^1$ is an* abstract term, *a term that represents a property.*

One basic assumption in term logic is that every term has a contrary. For example, the contrary of man is non-man and the contrary of wise is un-wise. The contrary of a term is expressed in $\mathcal{L}_T$ by negating that term. Thus, in language $\mathcal{L}_T$, we require a negation of terms as well as a negation of propositions. Our negation symbol is '$\sim$' which denotes 'not'. When the negation symbol prefixes a term, E, as in $\sim E$ and is read as 'un-E' or 'non-E'.

Definition 5.2. ***Negation of Terms***
n language $\mathcal{L}_T$, we extend the definition of terms from language $\mathcal{L}_V$ to include the following: If E^n is a term in $\mathcal{L}_T$, then $\sim E^n$ is also a term in $\mathcal{L}_T$. In particular, for any singular term e in $\mathcal{L}_T$, $\sim e$ is a term in $\mathcal{L}_T$, and for any abstract term E in $\mathcal{L}_T$, $\sim E$ is a term in $\mathcal{L}_T$. The term $\sim E^n$ is read as 'non-E^n' or 'un-E^n'

If I want to denote a *general term*, a term that is either negated or not negated (not both), I will use the Greek letters ϵ^n, δ^m, and θ^l. I only use general terms in key definitions and theorems. Otherwise, for the sake of simplicity, I mainly use the English letters (lower and upper case) E^n, F^m, and G^l for definitions and theorems where the substitution of negated terms may be assumed.

Definition 5.3. ***The Copula Verb***
The language $\mathcal{L}_T$ has a single binary verb letter V. For the binary verb V I give a special symbol: The single line, — , will replace the binary verb V. Intuitively, we view — as the copula verb, a relation of two terms. The symbol, — , is read as 'is'.

Definition 5.4. ***Negation of the Copula Verb***
if V is the copula, $\sim V$ is the negation *of V. In other words, the negation of the copula, — , is given by $\not-$, which is read as 'is not' or 'isn't'.*

In summary, the language $\mathcal{L}_T$ has singular terms e, f, g, terms that denote individuals, and abstract terms E, F, G, terms that denote properties, individual variables x, y, z, and the copula verb — . Also the language $\mathcal{L}_T$ contains the logical symbols $\rightarrow$ and $\sim$ and the quantifier symbol $\forall$.

Now we want to establish the formulas and propositions of $\mathcal{L}_T$. We begin with formation rules for $\mathcal{L}_T$ to form formulas, and then we define a proposition to be a formula with out any free variables. Recall by formation rule in $\mathcal{L}_V$, $V(E^n, F^m)$ is an atomic formula for $n, m = 0, 1$ where V is a binary verb. If V the copula, then we abbreviate $V(E^n, F^m)$ and write the following E^n— F^m. The formation rules of $\mathcal{L}_T$ are summarized below.

Definition 5.5. ***Formation Rules of*** $\mathcal{L}_T$

Let ϵ^n *and* δ^m *be (general) terms. By formation rules, the formulas in* $\mathcal{L}_T$ *are*

ϵ^n— δ^m, x— ϵ^n, *and* x— y, *where* n, m *= 0, 1 and* n *= 0 if* m *= 0.*

Note that the last condition, n *= 0 if* m *= 0, does not allow formulas such as* E — x *where* $n = 1$*; that is, it does not allow singular terms to occupy the object coordinate when an abstract term occupies the subject coordinate.*

Given any formulas ϕ *and* ψ, $\sim \phi$, $\phi \rightarrow \psi$, *and* $\forall x\phi$ *are formulas of* $\mathcal{L}_T$.

The language of $\mathcal{L}_T$ adopts the usual abbreviations for conjunction symbol $\wedge$ and the existential quantifier symbol $\exists$:$\phi \wedge \psi$ for $\sim (\phi \rightarrow \sim \psi)$ and $\exists x\phi$ for $\sim \forall x \sim \phi$.

Definition 5.6. ***Negation Rules***

- ***Negation Rule 1*** *(NR1). This rule relates verb negation with term negation. That is, the negative copula is equivalent to the positive copula with the object term negated. In symbols:*

  $$\not— E \text{ iff } — \sim E$$

  We may view this rule with the more general syntactic viewpoint:

  $$\sim V(N_l^r, N_m^s) \text{ iff } V(N_l^r, \sim N_m^s).$$

- ***Negation Rule 2*** *(NR2). For any terms* ϵ^n *and* δ^m *and the copula* — *, the negation for any atomic formula* ϵ^n— δ^m *is*

  $$\sim (\epsilon^n— \delta^m) \text{ iff } \epsilon^n \not— \delta^m \ (\sim (V(N_l^r, N_m^s)) \text{ iff} \sim V(N_l^r, N_m^s));$$

- ***Negation Rule 3*** *(NR3). Two negations adjacent to each other eliminate each other. In particular, for any term* E,

  $$\sim\sim E \text{ iff } E, \text{ and for any formula } \phi, \sim\sim \phi \text{ iff } \phi.$$

Propositions

A *sentence*, or *proposition*, is a formula which has no free variables. I use σ to denote a proposition. We now want the language $\mathcal{L}_T$ to provide elementary propositions that represent categorical propositions. There are four kinds of categorical propositions, and they are 'All S is P' (A form), 'No S is P' (E form), 'Some S is P' (I form), and 'Some S is not P' (O form), where S and P are abstract terms. To begin, we define what is called the simple universal statement and the simple existential statement.

The formula in $\mathcal{L}_T$, ϵ^n— δ^m where $n, m = 1$ is a relation of two properties. For example the formula may represent sentences such as 'Men are animals' or 'Men are wise', and these sentences imply or require quantification words 'all' or 'some' to clarify and make precise the intended assertion such as 'All men are animals' and 'Some men are wise'.

Therefore the formula ϵ^n— δ^m requires quantification to clarify meaning. Thus, ϵ^n— δ^m either is $\forall\, \epsilon^n$— δ^m or $\exists\, \epsilon^n$— δ^m. Definitions of $\forall\, \epsilon^n$— δ^m and $\exists\, \epsilon^n$— δ^m are given below.

Definition 5.7. ***Simple Universal Statement***

Let ϵ^n and δ^m be abstract terms ($n, m = 1$) that are either positive or negative. Let — be the copula verb, and let x be a variable. Then

$$\forall\, \epsilon^n \text{—}\, \delta^m \textit{ iff } \forall x\;\; x \text{—}\, \epsilon^n \rightarrow x \text{—}\, \delta^m,$$

where $\forall\, \epsilon^n$— δ^m is called the simple universal statement and is read as 'All ϵ^n is δ^m'.

Alternatively, we may view this definition syntactically:

$$\forall\, V(N_l^r, N_m^s) \textit{ iff } \forall x\, V(x, N_l^r) \rightarrow V(x, N_m^s),$$

where N_l^r, N_m^s are abstract noun symbols ($r, s = 1$), and V is the copula verb.

An example sentence in English of an elementary universal proposition is 'All men are Animals'. If we let the term M be the word 'men' and the term A be 'animals'. Then the following are equivalent: $\forall\;\; M — A$ iff $\forall x\;\; x — M \rightarrow x — A$,

Definition 5.8. ***Simple Existential Statement***

Let ϵ^n and δ^m be abstract terms ($n, m = 1$) that are either positive or negative. Let — be the copula verb, and let x be a variable. Then

$$\exists\, \epsilon^n \text{—}\, \delta^m \textit{ iff } \exists x\;\; x \text{—}\, \epsilon^n \wedge x \text{—}\, \delta^m,$$

where $\exists\, \epsilon^n$— δ^m is called the simple existential statement and is read as 'Some ϵ^n is δ^m'.

syntactically:

$$\exists\, V(N_l^r, N_m^s) \textit{ iff } \exists x\, V(x, N_l^r) \wedge V(x, N_m^s),$$

where N_l^r, N_m^s are abstract noun symbols, and V is the copula verb.

An example sentence in English of a simple existential statement is 'Some Animals are Men' or 'Some men are wise.' Let M be 'man' and W be 'wise'. Then the following are equivalent: $\exists\;\; M — W$ iff $\exists x\;\; x — M \wedge x — W$,

Note that the definitions of simple universal statement and simple existential statement correspond to equivalences brought about by the quantification equivalence (QE) $\exists x \phi$ iff $\sim \forall x \sim \phi$, the conjuction/conditional equivalence (CIE) $\phi \wedge \psi$ iff $\sim (\phi \rightarrow\sim \psi)$, negation rules NR1 and NR2, and the definitions Simple Universal Statement (SUS) and Simple Existential Statement (SES):

Let — be the copula and let E, F, be abstract terms, then we have the following equivalences:

$\forall\ E — F$	iff	$\forall x\ \ x — E \rightarrow x — F$	Def SUS
$\sim (\forall\ E — F)$	iff	$\sim (\forall x\ \ x — E \rightarrow x — F)$	apply negation
$\sim \forall\ E — F$	iff	$\sim \forall x\ \ x — E \rightarrow x — F$	
$\exists \sim (E — F)$	iff	$\exists x\ \sim (x — E \rightarrow x — F)$	QE
$\exists\ E \not— F$	iff	$\exists x\ \sim (x — E \rightarrow x — F)$	NR2
$\exists\ E — \sim F$	iff	$\exists x\ \ x — E \wedge \sim (x — F)$	NR1 and CIE
$\exists\ E — \sim F$	iff	$\exists x\ \ x — E \wedge x — \sim F$	NR2 of $x — F$

and by definition (SES) $\exists\ E — \sim F$ iff $\exists x\ \ x — E \wedge x — \sim F$. Also

$\exists\ E — F$	iff	$\exists x\ \ x — E \wedge x — F$	Def SES
$\sim \exists\ E — F$	iff	$\sim \exists x\ \ x — E \wedge x — F$	apply negation
$\forall \sim (E — F)$	iff	$\forall x\ \sim (x — E \wedge x — F)$	QE
$\forall\ E — \sim F$	iff	$\forall x\ \ x — E \rightarrow \sim (x — F)$	NR2 and CIE
$\forall\ E — \sim F$	iff	$\forall x\ \ x — E \rightarrow x — \sim F$	NR2 of $— F$

and by definition (SUS) $\forall\ E — \sim F$ iff $\forall x\ \ x — E \rightarrow x — \sim F$.

Definition 5.9 (**Elementary Propositions**). *Let ϵ^n and δ^m be general terms (terms that are either positive or negative). Then the elementary propositions in $\mathcal{L}_T$ are*

$$\epsilon^0 — \delta^0,\ \epsilon^0 — \delta^1,\ \forall\,\epsilon^1 — \delta^1,\text{ and } \exists\,\epsilon^1 — \delta^1.$$

In particular, let S, P be abstract terms, s be a singular term, then

$\forall\, S\ — P$ for 'All S is P'

$\exists\, S\ — P$ for 'Some S is P'

$\exists\, S\ \not— P$ for 'Some S is not P'

$\forall\, S\ — \sim P$ for 'All S is un-P'

$s — P$ for 's is P'

are elementary propositions in $\mathcal{L}_T$ that model the language of term logic.

5.2 Some Definitions of Term Logic

The Categorical Proposition

A *proposition* is a sentence that is either true or false. Recall in a formal calculus that a proposition is a formula in which has no free variables, and consequently a proposition is either true or false. I use σ to denote a proposition. In term logic, a proposition is made up of two terms connected in a relation by the copula; the two terms are the *subject term*, the term which occupies the subject of the verb, and the *predicate term*, the term which occupies the object of the verb. A *categorical proposition* is a proposition that asserts or denies the predicate term of (all or some) subject term. The act of affirming or denying something of the subject of a proposition is called *predication*. We will characterize the logical form of categorical propositions. The logical form of the categorical is specified with the notions of quality and quantity. These notions below are mainly from Sommers[16].

Quality (From Sommers) A *quality* is opposed as either positive or negative. Herein, oppositions of quality apply to terms, to the copula verb, and to predication.

Term Quality Any term is positive or negative with respect to another term which is called *term quality*. Term quality is related to the contrary of terms. Let E^n be a term where n = 0, 1. We represent the contrary of a term with the negation symbol $\sim$. If E is a term then *the contrary of* E is $\sim E$, where $\sim E$ is read as 'un-E' or 'non-E'. For example if E means 'wise' or 'man' then $\sim E$ means 'un-wise' or 'non-man'; and wise is logically contrary to unwise, and man is logically contrary to non-man.

We may call the term with the negation sign prefixed to the term, $\sim E$, a *negative term* and the term with out a negation sign prefixed to it, E, a *positive term*.

Verb Quality *Verb quality* is the positive or negative of a verb. If — is the copula, then — is the *positive copula* and $\not-$ is the *negative copula*, and we say that — and $\not-$ are contraries.

Herein, verb quality corresponds to what is known as predicate quality in Sommers. Predicate quality is the positive or negative of a predicate; that is, in the predicates 'is P' and 'isn't P' are quality opposites and the copulas 'is' and 'isn't' are correlatives. In $\mathcal{L}_\mathcal{T}$ this is expressed in the following symbols: for any term P, the predicate qualities are as — P and $\not-$ P where the copulas — and $\not-$ are correlatives.

The relation between term qualities and verb qualities may be illustrated in symbols of $\mathcal{L}_\mathcal{T}$. I will illustrate this in an example found in Sommers: Let 'ed' in the word 'coloured' to be a positive sign and the negative sign corresponding to 'less' in the word 'colourless'. The relation between the terms and the term quality is given by the classical laws which allows us to substitute 'isnt coloured' for 'is colourless' and 'is coloured' for 'isnt colourless'. The equivalent predicates 'isnt coloured' for 'is colourless' and 'is coloured' for 'isnt colourless' may be expressed may be expressed symbolically in $\mathcal{L}_\mathcal{T}$ as follows. Let C stand for 'coloured' and so $\sim C$ will stand for 'colourless', then by negation rule 1 'isnt coloured' for 'is colourless' is expressed as $\not-$ C for — $\sim C$, and 'is coloured' for 'isnt colourless' is expressed as — C for $\not-$ $\sim C$ by negation rules 1 and 3.

Term qualities and verb qualities are treated together as oppositions of contrariety. The logical contrariety of terms is E and $\sim E$, where E is any term; and the logical contrariety of verbs is — and $\not-$, which corresponds to term logics contrariety of predicates — E and $\not-$ E.

Let σ be an elementary proposition and let σ' be the elementary proposition whose terms are the same as σ and whose copula verb is the contrary of the copula in σ. Then σ and

σ' are called *contrary propositions*. For example, the proposition $\forall\, S\ \frac{\quad}{}\ P$ is contrary to $\forall\, S \not\frac{\quad}{} P$.

Predicative Quality There is a third qualitative opposition which Sommers makes use of and that is predicative quality. *Predicative quality* is the affirmation or denial of the proposition. To make use of this idea in $\mathcal{L}_T$, we assume a given set of propositions as affirmations, then we may apply the negation symbol to the propositions and obtain propositions as denials. It is clear that the negation symbol, $\sim$, applied to a proposition σ affects the predicative quality of σ, whether σ is affirmed or denied. That is, if σ is affirmed then $\sim \sigma$ is denied, and if σ is denied then $\sim \sigma$ is affirmed. We must keep in mind that the symbol $\sim$ prefixed to a proposition changes predicative quality and not confuse $\sim$ to be the sign which indicates affirmation or denial.

The concept of contradictory propositions in term logic is related to the concepts of affirmation and denial. For any proposition σ, σ and $\sim \sigma$ are *logical contradictories*.

Quantity (Sommers) The quantity of a proposition is either universal in quantity or particular in quantity. Let S and P be terms. In a proposition that has S as its subject term and P as its predicate term, *universal in quantity* is whenever P is affirmed of all S or denied of some S; and whenever a P is affirmed of some S or denied of all S is *particular in quantity*.

The Categorical Propositions Classically, there are four distinct kinds of categorical propositions, and they are determined by their quality and quantity. The four categorical propositions are described below.

Table 5.1. The Categorical Propositions and Language $\mathcal{L}_T$

Form	Term Logic notation	$\mathcal{L}_T$ notation
A	All S is P	$\forall\, S\ \frac{\quad}{}\ P$
E	No S is P (All S is not P)	$\forall\, S \not\frac{\quad}{} P$
I	Some S is P	$\exists\, S\ \frac{\quad}{}\ P$
O	Some S is not P	$\exists\, S \not\frac{\quad}{} P$

The table bellow, Table 5.2, displays equivalent forms for each categorical form. The equivalencies are easily verified by negation rules.

Note that for each category the first two equivalences are affirmations and the last two are denials.

We have completed the establishment of a formal language for term logic, $\mathcal{L}_T$, with the language $\mathcal{L}_V$ as its basis. We now want to establish a formal inference system for term logic. That is, we want to provide an inference system for the language $\mathcal{L}_T$, which is unique to term logic.

Table 5.2. Equivalent Forms of the Categorical Proposition and $\mathcal{L}_T$ Notation

Form	Term Logic notation	$\mathcal{L}_T$ notation
A	All S is P	$\forall\, S \;—\; P$
	All S is not un-P	$\forall\, S \not— \sim P$
	It is not the case that some S is not P	$\sim \exists\, S \not— P$
	It is not the case that some S is un-P	$\sim \exists\, S \;—\sim P$
E	No S is P (All S is not P)	$\forall\, S \not— P$
	All S is un-P	$\forall\, S \;—\sim P$
	It is not the case that some S is P	$\sim \exists\, S \;—\; P$
	It is not the case that some S is not un-P	$\sim \exists\, S \not— \sim P$
I	Some S is P	$\exists\, S \;—\; P$
	Some S is not un-P	$\exists\, S \not— \sim P$
	It is not the case that all S is not P	$\sim \forall\, S \not— P$
	It is not the case that all S is un-P	$\sim \forall\, S \;—\sim P$
O	Some S is not P	$\exists\, S \not— P$
	Some S is un-P	$\exists\, S \;—\sim P$
	It is not the case that all S is P	$\sim \forall\, S \;—\; P$
	It is not the case that all S is not un-P	$\sim \forall\, S \not— \sim P$

5.3 A Formal Inference System for Term Logic

My purpose in this section is to establish a formal theory that models term logic. Let T be a first-order theory with $\mathcal{L}_T$ as its underlying language that contains theory K_V above. In particular, theory T contains the inference system of K_V. Theory T will moreover include the following inference rules, which are based on the relation of terms.

Definition 5.10 (**Middle Term Rules**). *If two quantified simple propositions share a common term, called the middle term, then we may apply what are called middle term rules:*

- *Middle Term Rule 1 (MTR1)*
  $\forall\, \delta^m \;—\theta^l, \forall\, \epsilon^n \;—\delta^m \vdash_T \forall\, \epsilon^n \;—\theta^l$, *(where $n, m, l = 1$)*
- *Middle Term Rule 2 (MTR2)*
  $\forall\, \delta^m \;—\theta^l, \exists\, \epsilon^n \;—\delta^m \vdash_T \exists\, \epsilon^n \;—\theta^l$, *(where $n, m, l = 1$)*

With our two inference rules MTR1 and MTR2, we will derive all 24 valid syllogisms as theorems. Each of these theorems is named after the type of syllogism that the theorem is modeling.

Theorem 5.1. ***Barbara***

Let S, M, and P be abstract terms (n, m, l = 1). Then $\forall\, M \;—P, \forall\, S \;—M \vdash_T \forall\, S \;—P$.

Proof. Suppose we have the two propositions $\forall\, M\ — P, \forall\, S\ — M$. Then by MTR1 $\vdash_T \forall\, S\ — P$. □

This theorem is a specific syllogism of the first figure known as Barbara.

Theorem 5.2. ***Celarent***
Let S, M, and P be abstract terms. Then $\forall\, M\ \not— P, \forall\, S\ — M \vdash_T \forall\, S\ \not— P$.

Proof. Suppose we have the two propositions $\forall\, M\ \not— P$ and $\forall\, S\ — M$. By NR1 we have $\forall\, M\ —\sim P, \forall\, S\ — M$. Then by MTR1 we have $\vdash_T \forall\, S\ —\sim P$. Hence $\vdash_T \forall\, S\ \not— P$ by NR1. □

Hence we have as a theorem the syllogism of the first figure Celarent.

Theorem 5.3. ***Darli***
Let S, M, and P be abstract terms. Then $\forall\, M\ — P, \exists\, S\ — M \vdash_T \exists\, S\ — P$.

Proof. Suppose we have the two propositions $\forall\, M\ — P, \exists\, S\ — M$. Then by MTR2 $\vdash_T \exists\, S\ — P$. □

Thus we have the syllogism of the first figure Darli.

Theorem 5.4. ***Ferio***
Let S, M, and P be abstract terms. Then $\forall\, M\ \not— P, \exists\, S\ — M \vdash_T \exists\, S\ \not— P$.

Proof. Suppose we have the two propositions $\forall\, M\ \not— P, \exists\, S\ — M$. By NR1 $\forall\, M\ —\sim P$ and $\exists\, S\ — M$. Then by MTR2 $\vdash_T \exists\, S\ —\sim P$ and thus by NR1 $\vdash_T \exists\, S\ \not— P$. □

Thus we have the syllogism of the first figure Ferio.

We have derived as theorems all four valid syllogisms of the first figure. We now derive syllogisms of the second figure. In order to do so we need the following two lemmas.

Lemma 5.1. ***Universal Contraposition (UC)***
$\vdash_T \forall\, E\ — F$ iff $\vdash_T \forall\ \sim F\ —\sim E$

Proof

1	$\forall\, E\ — F$	Hyp
2	$\forall x\, x\ — E\ \to x\ — F$	1, DEF 4.8
3	$a\ — E\ \to a\ — F$	2, US
4	$\sim (a\ — F)\ \to \sim (a\ — E)$	3, A1 (Taut)
5	$\forall x\ \sim (x\ — F)\ \to \sim (x\ — E)$	4, UG
6	$\forall x\, x\ —\sim F\ \to x\ —\sim E$	5, NR1 and NR2
7	$\forall\ \sim F\ —\sim E$	6, DEF 4.8

Thus, by 1-7, we have $\forall\, E\ — F \vdash_T \forall\ \sim F\ —\sim E$. Hence $\vdash_T \forall\, E\ — F$ implies $\vdash_T \forall\ \sim F\ —\sim E$. The proof of the converse is similar.

Lemma 5.2. ***Existential Commutivity (EC)***
$\vdash_T \exists\, E\ — F$ *iff* $\vdash_T \exists\, F\ — E$

Proof

1	$\exists\, E\ — F$	Hyp
2	$\exists x\ x\ — E\ \wedge x\ — F$	1, DEF 4.9
3	$a\ — E\ \wedge a\ — F$	2, ES
4	$a\ — F\ \wedge a\ — E$	3, A1 (Taut)
5	$\exists x\ x\ — E\ \wedge x\ — F$	4, EG
6	$\forall x\ x\ — \sim F\ \rightarrow x\ — \sim E$	5, NR1 and NR2
7	$\exists\, F\ — E$	6, DEF 4.9

Thus, by 1-8, $\exists\, E\ — F \vdash_T \exists\, F\ — E$. Hence $\vdash_T \exists\, E\ — F$ implies $\vdash_T \exists\, F\ — E$. Proof of the converse is similar.

Theorem 5.5. ***Cesare***
Let S, M, and P be abstract terms. Then $\forall\, P\ \not— M, \forall\, S\ — M \vdash_T \forall\, S\ \not— P$.

Proof. Suppose we have the two propositions $\forall\, P\ \not— M, \forall\, S\ — M$. Then by NR1 $\forall\, P$ $— \sim M, \forall\, S\ — M$. By Lemma UC, $\forall\, M\ — \sim P, \forall\, S\ — M$. Then by MTR1, $\vdash_T \forall\, S$ $— \sim P$. Thus, by NR1 $\vdash_T \forall\, S\ \not— P$. Thus we have derived the syllogism of the second figure *Cesare*. □

Thus we have derived the syllogism of the second figure *Cesare*.

Theorem 5.6. ***Datisi***
Let S, M, and P be abstract terms. Then $\forall\, M\ — P, \exists\, M\ — S \vdash_T \exists\, S\ — P$.

Proof. Suppose we have the two propositions $\forall\, M\ — P, \exists\, M\ — S$ Then by Lemma EC, $\exists\, S$ $— M$. So by Darli we have $\exists\, S\ — P$. □

Thus we have derived the syllogism of the third figure Datisi.

The rest of the 24 traditional syllogisms are derived similarly.

Theory T is a formal theory for term logic. I call theory T a term calculus. A term calculus is a formal calculus that models term logic; more specific, a *term calculus* is a formal theory based on the language $\mathcal{L}_T$ without proper axioms.

In summary, the formal language $\mathcal{L}_V$ can be used to construct $\mathcal{L}_T$, which is a formal language for term logic. Hence $\mathcal{L}_V$ is a mathematical language that can model the natural syntax of term logic. The language $\mathcal{L}_V$ can also be used to construct $\mathcal{L}_P$, the language of predicate calculus by an abbreviation method for atomic formulas. Hence $\mathcal{L}_V$ is a mathematical basis for both formal languages $\mathcal{L}_P$ and $\mathcal{L}_T$. Thus, $\mathcal{L}_V$ is called a general formal language. Moreover, from the formal language $\mathcal{L}_P$ we can construct a predicate calculus, and from $\mathcal{L}_T$ we can construct a term calculus, like theory T. The language $\mathcal{L}_V$ is, therefore, a formal language that can be used to construct both a predicate calculus and a term calculus. The general formal language $\mathcal{L}_V$ is therefore a basis for a mathematical language, like predicate logic, and for a natural language, like term logic.

The following two sections of this chapter further exemplifies that the formal language can model the language of term logic, and consequently, natural syntax. They are sketches of how we can represent contemporary term logics as formal theories, or term calculi, with $\mathcal{L}_T$ as the underlying language. These sections do not provide any new conclusions, so they may be skipped by the reader who wishes to go straight to the theoretical implications of the general formal language.

5.4 TERM CALCULUS AND LUKASIEWICZ'S CALCULUS OF TERM LOGIC

In this section, I briefly go over how to establish Lukasiewicz's axiomatic method of Aristotel's logic as a formal theory with $\mathcal{L}_T$ as its under lying language. This will show that the formal language $\mathcal{L}_T$ can be used to model other deductive systems and thus theories of term logic.

We can formally express Lukasiewicz's term logic because we can express the language of his logic with the formal language $\mathcal{L}_T$. Lukasiewicz's axiomatic method of Aristotel's logic is a propositional calculus with some notions and rules that applies to terms and is given in his classic work, *Aristotle's Syllogistic from the Standpoint of Modern Formal Logic*. We can model these notions and rules with the language $\mathcal{L}_T$, while having a propositional calculus K_V. I will call this formal theory of Lukasiewicz's axiomatic method of Aristotel's logic theory L.

Let theory L contain theory T where $\mathcal{L}_T$ is its underlying language.

Following Lukasiewicz's deductive system[9], in theory L, we take the propositions A and I as primitives, $\forall\, A \mathrel{—} B$ and $\exists\, A \mathrel{—} B$, which are sentences in L. Lukasiewicz obtains the other two categorical propositions E and O by the following definitions:

D1. Proposition E: $\forall\, A \not\mathrel{—} B$ for $\sim (\exists\, A \mathrel{—} B)$, for any abstract terms A and B.

D2. Proposition O: $\exists\, A \not\mathrel{—} B$ for $\sim (\forall\, A \mathrel{—} B)$, for any abstract terms A and B.

In Theory L may account for D1 and D2 by establishing the equivalences $\forall\, A \not\mathrel{—} B$ for $\sim (\exists\, A \mathrel{—} B)$ and $\exists\, A \not\mathrel{—} B$ for $\sim (\forall\, A \mathrel{—} B)$ as theorems with the negation rules in theory T.

Theory L contains in addition to the logical axioms Λ, the following as proper axioms, which are the axioms of Lukasiewicz's deductive system:

1. $\forall\, A \mathrel{—} A$ 'All A is A'
2. $\exists\, A \mathrel{—} A$ 'Some A is A'
3. $\forall\, B \mathrel{—} A \wedge \forall\, C \mathrel{—} B \rightarrow \forall\, C \mathrel{—} A$ 'If all B is A and all C is B then all C is A.'
4. $\forall\, B \mathrel{—} A \wedge \exists\, B \mathrel{—} C \rightarrow \exists\, C \mathrel{—} A$ 'If all B is A and some B is C then some C is A.'

Axiom 1 is Aristotle's law of identity, and axioms 3 and 4 are the syllogisms Barbara and Datisi respectfully.

Lukasiewicz's has two rules inferences in his deductive system and they are rule of substitution and rule of detatchment (Modus Ponens).

Rules of Inference

(a) Rule of Substitution: If $\vdash_L \phi$ then if ψ is produced by ϕ by valid substitution then $\vdash_L \psi$. The only valid substitution is to substitute for terms (abstract terms) E, F, G other terms, e.g. F for E.

(b) Rule of Detatchment (Modus Ponens): If $\vdash_L \phi \rightarrow \psi$ and $\vdash_L \phi$ then $\vdash_L \psi$.

This completes the deductive system of Lukasiewicz's axiomatic method of Aristotel's term logic expressed as the formal theory L.

Lukasiewicz's, in addition to the two rules of inferences, makes use of 14 propositions which he proved as theorems to help deduce the valid syllogisms. These propositions are propositional tautologies in theory L, which are accounted for in axiom A1.

Theory L thus is equipped with the deductive appartus of Lukasiewicz's axiomatic system for Aristotel's logic. With the axioms and rules of inferences we can deduce the syllogisms as theorems by following Lukasiewicz's deductions. For the deductions of the valid syllogisms, I refer to Lukasiewicz. The deductions of the valid syllogisms in theory L follow in a analogous fashion. Theory L is thus a formal theory that models Lukasiewicz's axiomatic method of Aristotel's logic. Although theory L is a propositional calculus, it is based on the formal language of term logic, $\mathcal{L}_T$.

Hence we have formally modeled Lukasiewicz's axiomatic method of Aristotel's term logic by theory L which is based on the formal language $\mathcal{L}_T$. Hence Lukasiewicz's axiomatic method is modeled by a term calculus.

5.5 Term Calculus and Sommers' Calculus of Term

In this section I show how the formal language, $\mathcal{L}_T$, may be used as a basis for Sommers' calculus of terms. I provide an explicit one to one correspondence from the language of term calculus, $\mathcal{L}_T$, to the language of Sommers' calculus in his *Calculus of Terms*. Consequently, the notions of deduction in TC and Sommers calculus coincide in a one to one correspondence.

Sommers sometimes calls his calculus of terms the classical calculus, aluding to what Hobbes and Leibniz imagined how we naturally reason. So, I refer to his calculus as CC. I do not intend to give an exhaustive treatment of Sommers' calculus; interested readers are referred to Sommers' *A Calculus of Terms* [16]. I only give sufficient treatment of CC for the purpose of understanding what deduction is in his system and how to use his system, in particular how to test for validity.

In Sommers' calculus, CC, the primitive objects are the terms. In CC we let S be the subject term and P be the predicate term. Both terms are abstract terms (what I call *abstract terms* Sommers calls *universal terms*). Sommers represents the singular term as Si but herein we will represent the singular term with the lower case letter 's'. We do this so that the notation in TC and CC remains the same, and this allows us to see a direct correspondence of the language of CC with the language of TC.

The terms, namely S and P, are connected by the copula, the copula is either positive or negative. In CC the positive copula is represented by a plus sign '$+$', and the negative copula is represented by a minus sign '$-$'. Thus the following propositions may be formed in CC: $S + P$, which means 'S is P', and $S - P$, which means 'S is not P'.

We therefore have the following translations from languages of CC to the language of TC: $S + P$ iff S — P which means 'S is P'; and $S - P$ iff $S \not- P$ which means 'S is not P'.

The opposition of quantification in CC is also represented by the arithmetic signs '$+$' and '$-$'. The minus sign '$-$' represents the universal quantifier and the plus sign '$+$' represents the particular quantifier. Therefore, for our translation from the language of CC to the language of TC, we take $-$ iff $\forall$ and $+$ iff $\exists$. Now we can form the following propositions in CC with translations to TC: $-S + P$ iff $\forall\, S$ — P, which is read 'All S is P'; $-S - P$ iff $\forall\, S \not- P$, which is read 'All S is not P'; $+S + P$ iff $\exists\, S$ — P, which is read 'Some S is P'; $+S - P$ iff $\exists\, S \not- P$, which is read 'Some S is not P'.

Opposition of predication is also represented with plus and minus signs. The plus sign '$+$' represents affirmation, and the minus sign '$-$' represents denial. Thus, the proposition in the form of $-(-S + P)$ is read as 'it is not the case that all S are P'; $+(-S + P)$ is read as 'it is the case that all S are P'. Like in CC, we assume that the nonnegated elementary propositions in table 5.2 as affirmations and the negated propositions as denials.

Contrary of terms are represented with plus and minus signs. Let S be any term, then we say $(+S)$ is a positive term and $(-S)$ is a negative term being the contrary of $(+S)$. The negative term $(-S)$ is read as 'un-S'. So, the TC and CC correspondence is $\sim S$ iff $(-S)$. To simplify the notation, we assume S as a positive term; so S iff $(+S)$.

Below is a table which shows the notations of CC, TC, and term logic (TL) when depicting the categorical propositions A, E, I, and O:

The propositions in table 5.3 are what Sommers calls elementary propositions. The general form of an elementary proposition in CC is $\pm(\pm(\pm S) \pm (\pm P))$. To simplify the notation, if the proposition is an affirmative we drop the plus sign and remove the outer

Table 5.3. Notations of Sommers' Calculus and Term Calculus

Form	CC Notation	TC Notation	TL Notation
A	$+(-S+P)$	$\forall\, S\ —\, P$	All S is P
	$+(-S-(-P))$	$\forall\, S \not\!— \sim P$	All S is not un-P
	$-(+S-P)$	$\sim\exists\, S \not\!— P$	It is not the case that some S is not P
	$-(+S+(-P))$	$\sim\exists\, S\ — \sim P$	It is not the case that some S is un-P
E	$+(-S-P)$	$\forall\, S \not\!— P$	All S is not P (No S is P)
	$+(-S+(-P))$	$\forall\, S\ — \sim P$	All S is un-P
	$-(+S+P)$	$\sim\exists\, S\ — P$	It is not the case that some S is P
	$-(+S-(-P))$	$\sim\exists\, S \not\!— \sim P$	It is not the case that some S is not un-P
I	$+(+S+P)$	$\exists\, S\ — P$	Some S is P
	$+(+S-(-P))$	$\exists\, S \not\!— \sim P$	Some S is not un-P
	$-(-S-P)$	$\sim\forall\, S \not\!— P$	It is not the case that all S is not P
	$-(-S+(-P))$	$\sim\forall\, S\ — \sim P$	It is not the case that all S is un-P
O	$+(+S-P)$	$\exists\, S \not\!— P$	Some S is not P
	$+(+S+(-P))$	$\exists\, S\ — \sim P$	Some S is un-P
	$-(-S+P)$	$\sim\forall\, S\ — P$	It is not the case that all S is P
	$-(-S-(-P))$	$\sim\forall\, S \not\!— \sim P$	It is not the case that all S is not un-P

parenthesis as in $\pm(\pm S) \pm (\pm P)$. This completes the translations of elementary propositions which is a one to one correspondence of the languages of TC and CC.

Some basic inference rules of Sommers' calculus is given below which is taken directly from Sommers *Calculus of Terms*.

D.3 If two elementary statements have a common term, that term is said to mediate them and is called the *middle term* of the elementary statements. The two remaining terms are called the *extremes* of the statement formed by juxtaposition (conjunction) of the two elementary statements.

For example, in the conjunction of two elementary statements '$-S + M$ and $-M + P$' or 'All S is M and all M is P', M is the middle term, S and P are the extremes. Rule of conjunction in CC is provided later.

D.4 Two statements are called *similar* to one another if

(a) They are in the same scope; that is, they are either both universal or both particular.

(b) They have the same extremes.

D.5 A classical argument in standard form is an equation of two similar statements; on the left of the equality is an elementary statement, which is called the conclusion. In symbols let S and E be elementary statements then S = E is read 'S, hence E'. 'S = E' is called *immediate* if S is an elementary statement and *syllogistic* if S is a juxtaposition of elementary statements.

Thesis (I) 'S = E' is a valid argument if and only if S = E.

Now we are in the position to establish a correspondence between CCs notion of deduction with TCs notion of deduction. CC has two kinds of inferences, immediate and syllogistic. I will begin with the connection of immediate inference in CC with TCs equivalent concept of immediate inference.

Sommers Law of Immediate Inferences is below.

Law for Immediate Inference: If two statements and have the same logical quantifier (logical quantity), then ϕ entails ψ if and only if $\phi = \psi$. The equation '$\phi = \psi$' may be read as: ϕ, hence ψ.

Consider as example the contraposition of A Statements, from 'All S is P' to 'All un-P is un-S'. In CC this is expressed by the following algebraic statement $-S + P = -(-P) + (-S)$ and thus by simple algebra properties we see that equality holds and is therefore a valid inference: $-S + P = -(-P) + (-S) = P + (-S) = -S + P$. In TC, $-S + P$ is translated to $\forall\, S$ — P and then by universal contraposition we deduce $\forall\ \sim P$ — $\sim S$, which is translated to $-(-P) + (-S)$. Thus, in this case immediate inference in CC corresponds to a deduction in TC. Immediate inference in CC is deducible in both directions in TC. If we take the symbol $\vdash\dashv$ to stand for deducible in both directions, or dual deduction, then we may have a direct translation from CC to TC: $\vdash\dashv$ iff $=$ where the equality stands for immediate inference. Thus, we have a direct correspondence of the unpversal contraposition rule,

$$-S + P = -(-P) + (-S) \text{ iff } \forall\, S — P \vdash\dashv \forall\ \sim P — \sim S.$$

From the law above we can also say, with the same statements ϕ and ψ, that ψ entails ϕ if and only if $\psi = \phi$, and thus ψ, hence ϕ. Therefore this law in CC is the counterpart to the dual deduction of elementary statements with the same quantifier in TC. This is further illustrated in the following table. The table is based on Sommers' schedule of equations that covers all standard classical inferences that are immediate along with their corresponding duel deductions in TC.

Observe that the dual deductions $\vdash\dashv$ statements in table 5.4 are verified in TC by universal contraposition, particular conjunction, and negation rules.

Notice that the law of immediate inference does not cover inferences from universal to particular statements. Inferences from universal to particular fall under syllogistic inference. Before we go over the law of syllogistic inference, we need the notion of a juxtaposition of two elementary propositions. I will only go over a conjunction of elementary propositions, which will be enough for our purpose. Before this though we require the notion of nominalized terms.

Table 5.4. Immediate Inferences in Term Logic

	Form A
CC	$-S + P = -S - (-P) = -(-P) + (-S) = -(-P) - (+S)$
TC	$\forall\, S \text{---} P \vdash\dashv \forall\, S \not\text{---} \sim P \vdash\dashv \forall\, \sim P \text{---} \sim S \vdash\dashv \forall\, \sim P \not\text{---} S$
	Form E
CC	$-S - P = -S - (+P) = -P - S = -P - (+S)$
TC	$\forall\, S \not\text{---} P \vdash\dashv \forall\, S \not\text{---} P \vdash\dashv \forall\, \sim P \not\text{---} S \vdash\dashv \forall\, \sim P \not\text{---} S$
	Form I
CC	$+S + P = +S - (-P) = +P + S = +P - (-S)$
TC	$\exists\, S \text{---} P \vdash\dashv \exists\, S \not\text{---} \sim P \vdash\dashv \exists\, P \text{---} S \vdash\dashv \exists\, P \not\text{---} \sim S$
	Form O
CC	$+S - P = +S - (+P) = +(-P) + S = +(-P) - (-S)$
TC	$\exists\, S \not\text{---} P \vdash\dashv \exists\, S \not\text{---} P \vdash\dashv \exists\, \sim P \text{---} S \vdash\dashv \exists\, \sim P \not\text{---} \sim S$

Translations of Compound Propositions and Deduction

Nominalized terms are propositions that are treated as the terms of a compound proposition.

A compound statement is two or more elementary statements combined with a logical connective, e.g. the logical connective 'and' and 'if-then'. We will specifically go over conjunctions.

Let ϕ, ψ, and γ be elementary statements. Then, the conjunction of two propositions is $+[\phi] + [\psi]$, where $[\phi]$ and $[\phi]$ are nominalized terms; the compound proposition $+[\phi] + [\psi]$ is read as 'Some $[\phi]$ and $[\phi]$'. The conjunction of three elementary propositions is $+[+[\phi] + [\psi]] + [\gamma]$ where $[+[\phi] + [\psi]]$ and $[\gamma]$ are nominalized terms.

The conjunction in CC $+[\phi] + [\psi]$ is equivalent to $\phi \wedge \psi$, the conjunction in TC; and $+[+[\phi] + [\psi]] + [\gamma]$ is equivalent to $\phi \wedge \psi \wedge \gamma$. I establish this equivalence, the conjunction in CC and the conjunction in TC, with the following rule.

First, if one connective symbol is used repeatedly, we take grouping with parenthesis to the left. That is, $\phi_1 \wedge \phi_2 \wedge \phi_3$ is equivalent to $(\phi_1 \wedge \phi_2) \wedge \phi_3$, and $\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4$ is equivalent to $((\phi_1 \wedge \phi_2) \wedge \phi_3) \wedge \phi_4$, and so on.

Translation of conjunction statement in CC to conjunction statement in TC.

If we are given a conjunction in CC, we translate this to the conjunction in TC as follows:

1. omit the brackets around the elementary propositions and replace the remaining square brackets with parenthesis; that is, replace '[' with ')' and replace ']' with ')'.
2. Replace the right most plus sign with the conjunction symbol and eliminate the left most plus sign.

3. Repeat step 2 until all the plus signs are replaced by conjunctions or eliminated.

For example, if we want to translate $+[+[+[\phi_1] + [\phi_2]] + [\phi_3]] + [\phi_4]$ to the language of TC, we first apply rule 1 so that $+[+[+[\phi_1] + [\phi_2]] + [\phi_3]] + [\phi_4]$ becomes $+(+(+\phi_1 + \phi_2) + \phi_3) + \phi_4$. Then rule 2 gives $(+(+\phi_1 + \phi_2) + \phi_3) \wedge \phi_4$. Next apply of rule 2 again to get $((+\phi_1 + \phi_2) \wedge \phi_3) \wedge \phi_4$. Finally, one more application of rule 2 gives the desired result, $((\phi_1 \wedge \phi_2) \wedge \phi_3) \wedge \phi_4$. We may then drop the parenthesis since we assume left most grouping and have $\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4$.

Translation of conjunction statement in TC to conjunction statement in CC:

1. Begin with the conjunction statement in TC with full parenthesis; replace parenthesis with square brackets and apply square brackets around all elementary propositions.
2. For the left most closed brackets, replace the conjunction symbol with a plus sign and place a plus sign on the left of the left proposition.
3. Repeat step 2 until all conjunction signs are replaced by plus signs.

For example, given the compound proposition $((\phi_1 \wedge \phi_2) \wedge \phi_3) \wedge \phi_4$. rule 1 gives $[[[\phi_1] \wedge [\phi_2]] \wedge [\phi_3]] \wedge [\phi_4]$. Then rule 2 gives $[[+[\phi_1] + [\phi_2]] \wedge [\phi_3]] \wedge [\phi_4]$. Finally, applying rule 2 two more times gives our desired result $+[+[+[\phi_1] + [\phi_2]] + [\phi_3]] + [\phi_4]$.

Thus, $+[+[+[\phi_1] + [\phi_2]] + [\phi_3]] + [\phi_4]$ iff $((\phi_1 \wedge \phi_2) \wedge \phi_3) \wedge \phi_4$; $+[+[\phi_1] + [\phi_2]] + [\phi_3]$ iff $(\phi_1 \wedge \phi_2) \wedge \phi_3$; $+[\phi_1] + [\phi_2]$ iff $\phi_1 \wedge \phi_2$.

Unlike immediate inference, in syllogistic inference the equality sign is strictly a one way implication. So '=' in CC may corresponds $\vdash$ in TC. The example below is an illustration of syllogistic inference.

Example Barbara

Let S, M, and P be abstract terms. In the language of term logic the syllogism Barbara is 'All M is P and all S is M hence all S is P.' In the language of CC Barbara is expressed as

$$+[-M + P] + [-S + M] = +[-S + P],$$

and in the language of TC Barbara is expressed as

$$\forall\, M \,—\, P \wedge \forall\, S \,—\, M \vdash \forall\, S \,—\, P$$

which is a true deduction in theory T by its Barbara theorem.

Law for Syllogistic Inference: If the conjunction of two elementary statements ϕ and ψ equals an elementary statement γ which has the same extremes as ϕ and ψ, then equality is an entailment, a deduction. That is, ϕ and ψ deduces γ if and only if $+\phi + \psi = \gamma$. The equation '$+\phi + \psi = \gamma$' may be read as: ϕ and ψ, hence γ.

Below I provide deductions of CC, TC, and PC with an example taken from Sommers: 'Some clergymen are priests and all priests are bachelors and no bachelors are married hence some clergymen are not married.'

Let C be clergymen, P be priests, B be bachelors, and M be married.

CCs deduction

$+[+[+C + P] + [-P + B]] + [-B - M] = +C - M.$

If we assume +[+[+C + P] + [-P + B]] + [-B - M], then we proceed algebraically by first dropping the brackets and simplifying:

+[+[+C + P] + [-P + B]] + [-B - M] = C + P - P + B - B - M = C - M = +C - M.

TCs deduction

$\exists\, C — P \wedge \forall\, P — B \wedge \forall\, B \not— M \vdash \exists\, C \not— M.$

Assume $\exists\, C — P \wedge \forall\, P — B \wedge \forall\, B \not— M$. For $\exists\, C — P \wedge \forall\, P — B$, by MTR2 we deduce $\exists\, C — B$. So we have $\exists\, C — B \wedge \forall\, B \not— M$. By negation rule, we have $\exists\, C — B \wedge \forall\, B — \sim M$. By MTR2, $\exists\, C — \sim M$, and so by negation rule $\exists\, C \not— M$.

PCs deduction

$(\exists x\, x — C \wedge x — P) \wedge (\forall x\, x — P \rightarrow x —B) \wedge (\forall x\, x — B \rightarrow\sim (x —M)) \vdash \exists x\, x — C \wedge \sim (x — M)$.

Abbreviate to the monadic predicate relation:

$(\exists x\, Cx \wedge Px) \wedge (\forall x\, Px \rightarrow Bx) \wedge (\forall x\, Bx \rightarrow\sim Mx) \vdash \exists x\, Cx \wedge \sim Mx.$

Assume $(\exists x\, Cx \wedge Px) \wedge (\forall x\, Px \rightarrow Bx) \wedge (\forall x\, Bx \rightarrow\sim Mx)$. For $\exists x\, Cx \wedge Px$, by ES we have $Ca \wedge Pa$, where a is a constant. By conjunction, we have Ca and Pa. By US on $\forall x\, Px \rightarrow Bx$ we have $Pa \rightarrow Ba$. Because we have Pa and $Pa \rightarrow Ba$, by MP we have Ba. By US on $\forall x\, Bx \rightarrow\sim Mx$ we have $Bx \rightarrow\sim Mx$. By MP with Ba and $Bx \rightarrow\sim Mx$ we get $\sim Ma$. So by conjunction we have $Ca \wedge \sim Ma$. Hence, EG gives the desired result $\exists x\, Cx \wedge \sim Mx$.

I have shown that the language of TC has the ability to correspond with the language of CC, an established calculus for term logic. The language of TC therefore can include an algebraic as well as an axiomatic calculus.

CHAPTER 6

THEORETICAL IMPLICATIONS OF THE GENERAL FORMAL LANGUAGE

The general formal language, $\mathcal{L}_{\mathcal{V}}$, can apply to the language of mathematics and natural languages. The theoretical implications and applications of the general formal language are therefore found in the fields of mathematics and linguistics.

In this chapter, I discus some theoretical relations of the general formal language with mathematics and linguistics and then suggest approaches for investigating those relations. In mathematics, I suggest a way to construct a set theory with the language of term calculus, $\mathcal{L}_{\mathcal{T}}$, as its underlying language. This way begins with the mathematical construction of the copula as fundamental relations of set theory. In linguistics, I suggest a way to provide a deeper grammar for the atomic formula and thereby develop a grammar system for the general formal language, which can provide a mathematical basis for a grammatical theory and thus universal grammar.

6.1 THE LANGUAGE OF TERM CALCULUS AND SET THEORY, THE CONSTRUCTION OF THE COPULA AND THE FUNDAMENTAL RELATIONS OF SET THEORY

When examining kinds of meanings of the copula we find that the meanings of the copula are fundamental relations of set theory. Recall in chapter 2, the three distinct meanings of the copula are identity, membership, and attribution. So the copula as a relation may become the identity relation, the membership relation, or an attribution relation, depending on the terms of the verb. If the terms are both individuals, as in 'Clark Kent is superman', then the copula becomes the identity relation. If one term is an individual and the other is a property, as in 'Socrates is a man' or 'Socrates is wise', then the copula becomes equivalent to a membership relation. We may view membership as a special kind of attribution. Another special kind of attribution becomes obvious when the terms of the copula verb are classes. Observe the sentences 'Men are animals' and 'Primes are integers', where the copula as a relation expresses the subset relation. Thus, the meanings of the copula may be mathematically represented as equality, membership, and subset relations, especially when the terms of the copula represent either individuals or classes.

In this section, my purpose is provide a beginnings of a set theory with the language of term calculus as its underlying language. Specifically, I construct mathematical meanings of the copula for a formal theory, which I call theory S.

Let theory S, be an extension of theory T. As in theory T, this theory has one relation, the copula verb. The copula verb as a mathematical construction is seen when we limit our sense of ’property’. Herein, property is, syntactically, a noun or a noun phrase which represents, semantically, a quality, quantity, or class. I limit our sense of ’property’ to represent only a ’class’. Thus, in the language of S, the abstract terms denote classes. In general, I say that the terms represent *entities*, where an entity is either an individual or a class. These notions are treated in the following definition.

Definitions The symbols E^n, F^m, G^l are *entities* with rank n, m, and l, where $n, m, l = 0, 1, 2, 3, ...$ For the entity E^n the number n is called the *rank* of the, where n = 0, 1, 2,... If n = 0, then $E^n = E^0$ is an *individual*, so individuals have rank 0, we abbreviate e for E^0. Thus, the lower case letters, so e, f, g are individuals. If the rank is $n \geq 1$, then E^n is a *class*. Moreover, let E^n and F^m be properties, that is $n \geq 1$ and $m \geq 1$, then if $n > m$ we say that the property E has *a higher rank than property F*; in particular if $n > m$ we say that *class E has a higher rank than class F.* The capital letters with out superscripts E, F, G will be terms which denote classes; that is their rank is implied to be greater or equal to one.

Note that we extend the definition of rank *n* of an entity E^n from being equal to 0 or 1 to being equal to any nonnegative integer. The relation of rank of two classes as well as the relation of rank of individual and class may be intuitively grasped in the following mathematical statement:

$$2 \in \mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R}$$

where $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$ are the sets of natural numbers, integers, rationals, and reals, respectively. In the above statement, 2 is taken as an individual so it has rank 0; the classes $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$ all have rank greater or equal to 1; and, for instance, $\mathbb{N}$ has a lower rank than $\mathbb{Z}$, being a proper subset. That is, for any classes A and B, if A is the proper inclusion of B, then A has a lower rank than B. This sense of rank is captured in the definition of proper inclusion in theory S below.

Let S be a theory that contains theory T. Thus $\mathcal{L}_T$ is the underlying language of S. The language of S contains, in general, entities E^n, F^m, G^l,..., where n, m, and l are nonnegative integers which represent the ranks of the entities. If the ranks are equal to zero, then the entities are individuals which are represented as the lower case letters $e, f, g, ...$ If the ranks of the entities are greater or equal to one, then the entities represent classes which are represented by the upper case letters $E, F, G, ...$ The language of S has a single binary verb symbol V, which is the copula verb. The copula verb is denoted as — .

Formation Rule in S. If E^n and F^m are any entities and — is the copula verb, then the elementary formulas of the language of S is of the form E^n — F^m where $n, m = 0, 1, 2, ...$

Recall in theory T the formula E^n — F^m is read as 'E^n is F^m'. In theory S we specify the meanings of copula. The meanings of the copula is dependent on the relation of rank between the two entities under the copula relation. The different relations of rank determine the different meanings of the copula, all of which are set theoretic relations. One of the meanings of the copula is equality, which is defined as follows.

Definition 6.1. ***Equality***

For any entities E^n and F^m, then

$$E^n = F^m \text{ iff } E^n \text{ — } F^m \text{ where } n = m.$$

Thus, we established equality from a verb (or verb symbol). This is similar to establishing equality from a predicate symbol in predicate calculus. We may therefore adopt the usual axioms for equality and derive the expected properties of equality. These axioms of equality are specifically A5 and A6, which are axioms in theory K_V (chapter 3). Recall that theory T contains theory K_V, so theory S contains K_V. From these axioms we may derive the three main properties of equality, reflexive, symmetric, and transitive. Proof of these properties, which use A5 and A6, are found in Mendelsons *Introduction to Mathematical Logic*. The properties of equality are summarized in the following theorem.

Theorem 6.1. ***Properties of Equality***

$\vdash_S E^n = E^n$

$\vdash_S E^n = F^m \rightarrow F^m = E^n$

$\vdash_S E^n = F^m \wedge F^m = G^l \rightarrow E^n = G^l$

Hence S is a first order theory with equality.

Thus the copula — determines equality when the ranks of the two terms under the verb are equal. The copula determines another fundamental relation of set theory and mathematics and that is the membership relation.

Definition 6.2. ***Membership Relation***

The membership relation is denoted by $\in$. For any entities E^n and F^m with $0 = n < m$. then

$$e \in F^m \text{ iff } E^n \text{ — } F^m \text{ where } 0 = n < m,$$

where $e \in F^m$ means 'the individual, e, is a member of the class, F.'

Note that, because of the condition $0 = n < m$, the membership relation is antisymmetric. That is, we cannot have both $E \in F$ and $F \in E$. This moreover shows that $E \in E$ cannot be formed, which means that theory S avoids Russel's paradox. Russel's Paradox: Let $R = \{x | x \notin x\}$, then $R \in R$ iff $R \notin R$.

Observe that the relation of rank of two classes and the order of the two classes under the copula verb imply certain quantification in the following simple sentences:

(All) Primes are Integers	(All) Men are Animals.
(Some) Integers are primes	(Some) Animals are men.

The observation above motivates the following definition.

Definition 6.3 (Relation of Rank and Quantification). *Let E^n, F^m be classes ($1 \leq n, m$). Then*

- E^n — F^m *where* $n \leq m$ *iff* $\forall\, E^n$ — F^m,
- E^n — F^m *where* $n > m$ *iff* $\exists\, E^n$ — F^m.

The following definition is the simple universal statement found in theory T.

Definition 6.4. ***Simple Universal Statement***

Let x be an individual variable and let E^n and F^m be classes ($n, m \geq 1$). Then

$$\forall\, E^n\text{—}\, F^m \text{ iff } \forall x \;\; x \text{ — } E^n \rightarrow x \text{ — } F^m \text{ iff } \forall x \;\; x \in E^n \rightarrow x \in F^m,$$

by definition of membership.

Hence, E^n — F^m where $1 \leq n \leq m$ iff $\forall x \;\; x \in E^n \rightarrow x \in F^m$.

The next definition is where we apply the connection of the strict inequality of rank of two classes and proper inclusion. I define inclusion differently from the common definition found in set theory. The common definition found in a set theory is $E^n \subseteq F^m$ iff $\forall x\, x \in E^n \rightarrow x \in F^m$. I instead prove this equivalence in a following theorem.

Definition 6.5. ***Proper Inclusion***

Let E^n, F^m be classes. Then

$$E^n \text{ — } F^m, \text{ where } 1 \leq n < m, \text{ iff } E^n \subset F^m.$$

Definition 6.6. ***Inclusion***

$$E^n \subseteq F^m \text{ iff } E^n \subset F^m \text{ or } E^n = F^m.$$

Now we are ready to establish the common definition of inclusion as a theorem. We will also establish the common mathematical identity $E^n = F^m$ if and only if $E^n \subseteq F^m$ and $F^m \subseteq E^n$ as a theorem.

Theorem 6.2. ***Inclusion***

Let E^n and F^m be classes. Then

$$E^n \text{ — } F^m \text{ where } 1 \leq n \leq m \text{ if and only if } E^n \subseteq F^m.$$

Proof. Assume E^n — F^m where $1 \leq n \leq m$. Now $n \leq m$ means $n = m$ or $n < m$. If $n = m$ then $E^n = F^m$ by Definition 6.1. If $n < m$ then $E^n \subset F^m$ by Definition 6.5. Hence we have $E^n \subset F^m$ or $E^n = F^m$. Thus, by Definition 6.6 $F^m \subseteq E^n$. Conversely, assume $F^m \subseteq E^n$. This means by Definition 6.6 that $E^n \subset F^m$ or $E^n = F^m$. If $E^n \subset F^m$ then E^n — F^m, where $1 \leq n < m$, by Definition 6.5, and if $E^n = F^m$ then E^n — F^m where $n = m$ by Definition 6.1. Hence we have E^n — F^m with $1 \leq n < m$ and E^n — F^m with $n = m$. Therefore, E^n — F^m where $1 \leq n \leq m$. □

Theorem 6.3. *Let E^n and F^m be classes. Then*
$F^m \subseteq E^n$ iff $\forall x \;\; x$ — $E^n \rightarrow x$ — F^m where $1 \leq n \leq m$.

Proof. $E^n \subseteq F^m$ iff E^n — F^m where $1 \leq n \leq m$ by Theorem 6.2 iff $\forall x \;\; x \in E^n \rightarrow x \in F^m$ by Definition 6.4. □

Theorem 6.4. *Let E^n and F^m be classes. Then*

$$E^n = F^m \text{ if and only if } E^n \subseteq F^m \text{ and } F^m \subseteq E^n.$$

Proof. Assume $E^n = F^m$ with $n, m \leq 1$. By Definition 6.1 we have E^n — F^m with $1 \leq n = m$. This means E^n — F^m where $1 \leq n \leq m$ and $1 \leq m \leq n$. Hence by Theorem 6.4 we have $E^n \subseteq F^m$ and $F^m \subseteq E^n$. Conversely, assume $E^n \subseteq F^m$ and $F^m \subseteq E^n$. Hence by Theorem 6.2 we have E^n — F^m with $1 \leq n \leq m$ and F^m — E^n with $1 \leq m \leq n$. Because $n \leq m$ and $m \leq n$ implies $n = m$, therefore E^n — F^m with $1 \leq n = m$. Hence by Definition 6.1 $E^n = F^m$. □

Now I wish to establish what the copula means when the subject term of the classes under the copula, has a higher rank than the object term. That is, what the copula means for E^n — F^m with $n > m$ where $m, n = 1, 2, ...$ E^n — F^m where $n > m$ iff $\exists$ E^n — F^m.

We therefore let the sentence F^m— E^n with $m > n$ and $m, n = 1, 2, ...$, to be the quantified sentence $\exists$ F^m— E^n, since the copula in $\exists$ F^m— E^n can be expressed a set theoretic relation. The meaning of the copula for E^n — F^m where $n > m$ $(m, n = 1, 2, ...)$ is given below in the following three definitions, beginning with the defintition of the simple existential statement found in theory T.

Definition 6.7. ***Simple Existential Statement***
Let x be an individual variable and let E^n and F^m be classes $(n, m > 1)$, and — be the copula verb. Then

$$\exists\, E^n\text{— } F^m \text{ iff } \exists x \;\; x \text{ — } E^n \wedge x \text{ — } F^m \text{ iff } \exists x \;\; x \in E^n \wedge x \in F^m.$$

Hence, E^n — F^m where $1 \leq m < n$ iff $\exists x \;\; x \in E^n \wedge x \in F^m$.

Definition 6.8. ***Empty Set***

Let $\emptyset$ be the empty set. The rank of the empty set is assumed to be zero.

Definition 6.9. ***Set Existential Intersection***

Let x be a variable and let E^n and F^m be classes, where $0 < n < m$. Then

$$E^n \bar{\cap} F^m \text{ if and only if } \exists\ E^n — F^m,$$

where $E^n \bar{\cap} F^m$ means $E^n \cap F^m \neq \emptyset$.

Theorem 6.5. *Let E^n and F^m be classes. Then*

$$E^n — F^m\ n > m \text{ iff } E^n \bar{\cap} F^m.$$

Proof. $E^n — F^m\ n > m$ iff $\exists x\ \ x \in E^n \wedge x \in F^m$ iff $E^n \bar{\cap} F^m$. □

The meanings of the copula verb, — , are summarized below.

Let E^n and F^m be entities and $E^n — F^m$ be a sentence in S. Then

$$\begin{array}{lll} n = m & \text{iff} & E^n = F^m, \\ 0 = n < m & \text{iff} & E^n \in F^m, \\ 0 < n \leq m & \text{iff} & E^n \subseteq F^m, \\ 0 < n < m & \text{iff} & E^n \subset F^m, \\ 0 < m < n & \text{iff} & E^n \bar{\cap} F^m. \end{array}$$

The set relations are mathematical meanings for the copula. These mathematical meanings have a connection with the meanings for the copula in a natural language such as English. The table below exemplifies this.

Table 6.1. The Meanings of the Copula as Set Theoretic Relations

$e \in F$	Socrates is a man.	2 is an integer.
$E = F$	Hesperus is Phosphorus.	x is 2.
$E \subset F$	All men are animals	All primes are integers.
$F \bar{\cap} E$	Some animals are men	Some integers are primes

This completes our construction of the copula. The meanings of the copula are fundamental set theoretic relations $=$, $\in$, $\subset$, and $\bar{\cap}$, and they are determined by the relation of the ranks of the two entities under the copula. These meanings have a correspondence with the meanings of the copula in English, which shows a connection between the language of mathematics and natural languages.

We may also say that the fundamental relations of set theory are reduced to a single relation, the copula. This reduction of the set theoretical relations may have foundational implications.

Theory S may serve as a beginning for a set theory with the language of term calculus, $\mathcal{L}_{\mathcal{T}}$, as its underlying language. The usual underlying language for set theory is the language of predicate calculus. We know that the language of TC has more structure, being more fundamental or general, than the language of PC. Thus a set theory with TC as its underlying language may reveal general or even foundational ideas in set theory. Because set theory is used as a foundation of mathematics, a fully established set theory with TC as its underlying language may shed light in the foundations of mathematics.

6.2 Mathematical Grammar, The General Formal Language and the Grammar System

I view that the general semantic representation of a simple sentence in a natural language, as well as a mathematical language, is a relation. The general formal language, $\mathcal{L}_{\mathcal{V}}$, is based on the idea that in a sentence a verb is the syntax of a relation. I believe that the idea of the verb as a relation is the natural syntactic structure of a sentence. In the table below, observe the following kinds of simple sentences in English and their representations as verb relations.

Table 6.2. Kinds of Simple Sentences in English with the Verb as a Relation

Simple Sentence	Verb as the Syntax of a Relation	In Grammatical Relation Symbols
Bogart kisses Bacall.	kisses(Bogart, Bacall)	$V(S, O)$
Hudini disapeared.	disapeared(Hudini)	$V(S)$
Plato walks.	Walks(Plato)	$V(S)$
Hesperus is Phosphorus.	is(Hesperus, Phosphorus)	$V(S, O)$
Socrates is a man.	is (Socrates, a man)	$V(S, O)$
Charity is beautiful.	is(Charity, beautiful)	$V(S, O)$
John gave Mary a flower.	gave(John, a flower, Mary)	$V(S, O_d, O_i)$

From observing the table above, we may wonder whether there is a relation between the kind of verb and the kind of sentence or relation. We may ask, whether the verb not only determines syntactic structure with valency but also a semantic one as well. Questions like this eventually leads one to imagine a grammar system for the general formal language. If the idea of a verb as a relation is indeed in accordance to natural syntax, the general formal language along with a grammar system may serve as a mathematical framework for a theory of grammar. A grammar system for $\mathcal{L}_{\mathcal{V}}$ is a system of definitions and rules that form (or generate) all the kinds of grammatical sentences in a natural language and rules that explain grammatical phenomenon.

The grammar system is a system that is based on the language $\mathcal{L}_{\mathcal{V}}$ and provides a deeper syntactic and semantic structure for atomic formulas and thus sentences. A grammar system, for example, will define kinds of verbs like transitive versus intransitive verbs, provide semantic roles like agent and patient for the grammatical relations subject and object, provide rules like passivity of action verbs, etc. I call the general formal language, $\mathcal{L}_{\mathcal{V}}$, together with a grammar system a mathematical grammar (MG). If a grammar system is developed, I envision that MG may be a mathematical basis for a theory of grammar and thus universal grammar. If MG is a mathematical basis for universal grammar, which is the study of principles of grammar universal to all languages, MG will shed light on the foundations of natural language.

My purpose is to give a brief on how MG may be developed and applied as a mathematical basis for current grammatical theory in the field of linguistics, namely generative grammar, dependency grammar, and relational grammar. I do not establish a grammar system and thus MG. I rather sketch ideas of possible approaches to the development of MG as it relates to current grammatical theory.

6.2.1 Mathematical Grammar and Generative Grammar

Recall that the language of semantics is a metalanguage, which is some natural language. We therefore begin the development of the grammar system with the interpretation, or structure, $\mathfrak{G}$ of the language $\mathcal{L}_V$. The structure $\mathfrak{G}$ is defined as $\mathfrak{G} = (\mathbf{N}, \mathscr{N}, \mathbf{V})$, where **N** is the universe, the set of individuals, which is also the set of concrete nouns of a natural language; $\mathscr{N}$ is a set of subsets of **N** ($\mathscr{N}$ is a family of sets), the set of properties, which is also the set of abstract nouns of a natural language; and **V** is the set of relations on any cartesian combination of **N** and $\mathscr{N}$, so **V** the set of relations on $\mathscr{N}$, on $\mathbf{N}\times\mathscr{N}$, etc; **V** is also the set of verbs of a natural language. We therefore have $\mathfrak{G}$ as a basic lexicon of a natural language, since, in linguistics, a lexicon is an inventory of meanings, or denotations, of a natural language.

We may add more structure to the lexicon by including adjectives and adverbs, and determiners in our language. This will increase the capacity of MG to produce a greater variety of grammatical sentences for a natural language.

Let the language $\mathcal{L}_V$ contain, in addition to noun symbols N and verb symbols V, adjective symbols Adj, adverb symbols Adv, preposition symbols P, and determiner symbols Det. Then with our interpretation $\mathfrak{G}$ of language $\mathcal{L}_V$, we provide the semantics for the symbols as follows. In the meta-language, let $\mathbf{D_{ET}}$ be the set of determiners, $\mathbf{A_{DJ}}$ be the set of adjectives, $\mathbf{A_{DV}}$ be the set of adverbs, **P** be the set of prepositions. Then we further define $\mathfrak{G}$ as follows:

$\mathfrak{G}(Det) \in \mathbf{D_{ET}}$, $\mathfrak{G}(Adj) \in \mathbf{A_{DJ}}$, $\mathfrak{G}(Adv) \in \mathbf{A_{DV}}$, $\mathfrak{G}(P) \in \mathbf{P}$.

Hence our structure $\mathfrak{G}$ of a language $\mathcal{L}_V$ is $\mathfrak{G} = (\mathbf{N}, \mathscr{N}, \mathbf{V}, \mathbf{A_{DJ}}, \mathbf{A_{DV}}, \mathbf{D_{ET}}, \mathbf{P})$, which models the lexicon of a natural language.

Let N = $\mathfrak{G}(N_l^r)$ be a noun of a natural language, let V = $\mathfrak{G}(V) \in \mathbf{V}$ be a verb of a natural language, Adj = $\mathfrak{G}(Adj) \in \mathbf{A_{DJ}}$ be an adjective of a natural language, etc. Now we are in the position to provide more grammatical structure for the atomic formula with phrase structure rules. In MG, the phrase structures NP, VP, and PP, transform to more complex structures. NP transforms to the structure Det-Adj-N, i.e. NP $\rightarrow$ Det-Adj-N, where Det is a determiner, Adj is an adjective, and N is a noun. PP transforms to the structure P-NP, i.e. P $\rightarrow$ PP-NP, where P is a preposition. VP transforms to the structure Aux-Adv-V, i.e. VP $\rightarrow$ Aux-Adv-V, where Aux is an auxiliary verb, Adv is an adverb, and V is a verb.

These phrase structure rules provide more structure to the atomic formula. With phrase structure rules, we may express grammatical simple statements which have, for instance, the three component form *noun phrase-verb phrase-noun phrase*, in symbols NP-VP-NP, where NP denotes a Noun Phrase and VP denotes a Verb Phrase, where the VP is a relation on NPs. We may also express grammatical simple statements which have the three component structure *noun phrase-verb phrase-prepositional phrase*, in symbols NP-VP-PP, where PP denotes a Prepositional Phrase, where the VP is a relation of NP and PP. We may, further, express simple statements which have four component structure NP-VP-NP-PP for ditransitive verbs. The atomic formulas therefore become, for instance, VP(NP, NP), VP(NP, PP), VP(NP, NP, PP), which are general representations of simple sentences. This also generalizes the notion of a verb as a relation of nouns to the notion of the verb phrase as a relation of noun phrases.

Now lets explore an example how MG accounts for the sentence *The small boy swiftly hits the ball*.

By formation rule we have VP(NP_l^r, NP_m^s), then apply the interpretation $\mathfrak{G}$ so we have

$$\begin{aligned}
\mathfrak{G}(\text{VP}(\text{NP}_l^r, \text{NP}_m^s)) &= \mathfrak{G}(\text{VP})(\mathfrak{G}(\text{NP}_l^r), \mathfrak{G}(\text{NP}_m^s)) \\
&= \mathfrak{G}(adv\ V)(\mathfrak{G}(det\ adj\ N_l^r),\ \mathfrak{G}(det\ N_m^s) \\
&= \mathfrak{G}(adv)\ \mathfrak{G}(V)(\mathfrak{G}(det)\ \mathfrak{G}(adj)\ \mathfrak{G}(N_l^r)),\ \mathfrak{G}(det)\ \mathfrak{G}(N_m^s)) \\
&= \mathfrak{G}(det)\ \mathfrak{G}(adj)\ \mathfrak{G}(N_l^r)\ -\ \mathfrak{G}(adv)\ \mathfrak{G}(V) -\ \mathfrak{G}(det)\ \mathfrak{G}(N_m^s) \\
&= \textit{The small boy swiftly hits the ball}
\end{aligned}$$

Phrase structure rules originated in generative grammar. Generative grammar (GG) is the standard linguistic theory of grammar. The phrase structure rules, or production rules, in MG differs from the standard model of generative grammar, GG. A sentence in GG begins with the a sentence S which starts the production by transforming to NP-VP, where VP, the verb phrase, transforms to Aux-Adv-V-NP. In symbols, S$\rightarrow$NP-VP and VP$\rightarrow$Aux-Adv-V-NP.

The phrase structure VP in GG is more of a predicate phrase structure, since it transforms to a noun phrase along with a verb phrase. This is because of sentence formation in GG is based on the traditional subject-predicate distinction. In MG the phrase structure VP differs from that in GG; in MG, VP does not contain a noun phrase NP. This is because sentence formation in MG is based on a subject-verb-object distinction, where the verb is a relation of subject and object.

MG may be a basis for GG. GG, however, must adopt the idea of a verb as a relation and the subject-verb-object distinction as opposed to the subject-predicate distinction as a basis for sentence formation. We may then further furnish the grammatical system with the methods and findings of GG. For example, in order to generate more types of sentences, MG may adopt the notion and use of transformations in the grammatical system from generative grammars. A theory of transformations in the grammatical system would increase MGs capacity to produce a greater variety of sentences like embedding and sentences with adverbial clauses.

6.2.2 Mathematical Grammar and Dependency Grammar

In MG, the arity of the verb is based on the notion of valency in linguistics. The *valency* of the verb is ”the capacity a verb has for combining with particular patterns of other sentence constituients, in a similar way to that in which the valency of a chemical element is its capacity for combining with a fixed number of atoms of another element.[1]” Valency is the (least) number of arguments required to make a grammatical sentence. The arguments are the grammatical relations, subject and object(s) of the verb. Lucien Tesnire originally developed the idea of verb valency in his theory of grammar, dependency grammar. Dependency grammar takes the verb to be the structural center of the sentence, where the verb is the root of all sentence structure. All other syntactic constituents such as nouns, adjectives, etc., are either directly or indirectly connected to the verb by directed links called *dependencies* [1].

Dependency grammar may offer contributions to the grammar system like, for example, methods of increasing or decreasing the valency of the verb. In $\mathcal{L}_{\mathcal{V}}$, we form what are called core valency sentences. The grammar system may incorporate the methods of dependency grammar to increase or decrease the valency of the verb. For example, we may increase the valency of the verb ’walks’ to include an indirect object or prepositional phrase so we may form ’Plato walks in the forest’.

Valency theory and other aspects of dependency grammar may increase MGs sentence production capacity. Dependency grammar is a verb central theory which accords well with the idea of the verb as a relation. Dependency grammar may offer an altogether different approach to realize MG than the approach which uses the methods of GG. This approach, if successful, will provide a mathematical basis for dependency grammar.

6.2.3 Mathematical Grammar and Relational Grammar

Besides being a verb central theory, MG mathematically models the grammatical relations. The grammatical relations are the subject, verb, and objects of a sentence. The idea of the verb as a relation consequently exposes the the grammatical relations of a sentence. If the verb V has valency 2, then V is a relation of subject, S, and object, O, i.e. as a relation in symbolic form V(S, O) or S-V-O. Note that the subject S and the object O are noun phrases, NP. Furthermore because the relation is ordered, we can define that the subject of a relation is in the first coordinate of the ordered pair and the object is in the second coordinate of the ordered pair. Thus, the verb is a relation and the order underlying the relation defines the subject and object.

If V has valency 3, then the sentence as a relation is V(S, Od, Oi) where Od is the direct object and Oi is the indirect object. Note that Od is a noun phrase (NP) and Oi is a prepositional phrase (PP).

A grammatical theory that takes the grammatical relations as primitive objects is relational grammar. Relational grammar gives theoretical primacy to the grammatical relations believing that a good deal of grammatical phenomena is better analyzed directly in terms of grammatical relations [11].

In Relational Grammar the grammatical relations, the subject and objects of the verb, are represented by numbers. The subject by (1), direct object by (2) and indirect object by (3) [8]. For example, a sentence in RG is the following:

1	P	3	2
John	gave	Mary	a flower

where P is a predicate defined in the modern sense.

Observe that the numbering scheme in RG corresponds to the coordinate number of subject and objects as arguments of the verb in MG: gave(John, a flower, (to) Mary), where ’John’ is in the 1st coordinate, ’a flower’ is in the second coordinate, and ’Mary’ is in the third coordinate of the verb relation ’gave’.

This correspondence suggests that MG may be an ideal mathematical basis for the grammatical theory RG.

Relational grammar argues that the grammatical relations provide the ideal means to state syntactic rules in universal terms. Grammatical relations are considered universal since they seem to exist in every human language, and further the languages that have grammatical relations, the grammatical relations play the same role in every language [18].

The numbered roles above are used to state a set of linguistic universals which makes RG a theory for universal grammar. Therefore, if MG is indeed a mathematical basis for RG, then MG is a mathematical basis for universal grammar.

CHAPTER 7
CONCLUSION

The verb on nouns as the syntactic structure for atomic formulas provides a more general structure for the atomic formula than the predicate on terms syntactic structure. It reveals a deeper grammatical structure for the atomic formula and expands the sense of terms in a mathematical language where terms may either denote an individual or property.

I constructed a formal language that has atomic formulas with a verb on nouns syntactic structure. I showed that this formal language, $\mathcal{L}_V$, can be used to construct the formal language of predicate calculus, $\mathcal{L}_P$. By a method of abbreviating atomic formulas with a verb on nouns structure to a predicate on terms structure (where terms are limited to represent individuals), the formal language, $\mathcal{L}_V$ has atomic formulas with a predicate on terms syntax, and thus becomes $\mathcal{L}_P$. From the formal $\mathcal{L}_P$ we can construct a predicate calculus. Hence $\mathcal{L}_V$ is a basis for $\mathcal{L}_P$ and thus for predicate calculus.

I also showed that the mathematical language, $\mathcal{L}_V$, can model natural syntax. I applied the language to the syntax of term logic, which is based on natural syntax, by furthering its construction to the formal language, $\mathcal{L}_T$, a formal language for term logic. From $\mathcal{L}_T$ I constructed a term calculus, a formal calculus for term logic.

Because $\mathcal{L}_V$ is a basis for both formal languages $\mathcal{L}_P$ and $\mathcal{L}_T$, $\mathcal{L}_V$ is a general formal language. Language $\mathcal{L}_V$ can be used to construct both a predicate calculus and a term calculus. Hence the general formal language, $\mathcal{L}_V$, mathematically rigorous for a predicate calculus yet natural enough for a term calculus.

The general formal language, being a basis for natural language and the language of mathematics. must have theoretical implications and applications in the fields of mathematics and linguistics. In mathematics, I provide the beginnings of the construction of a set theory based on the language of term calculus. In this set theory, I derived the fundamental relations of set theory from the copula verb, i.e. the copula verb is a reduction of the fundamental relations equality, membership, and inclusion.equality, membership, and inclusion. Since set theory is applied in the foundations of mathematics, a fully established set theory with the language of term calculus as its underlying language may shed light in the foundations of mathematics.

In linguistics, I provide a sketch of a grammar system for the general formal language. The general formal language with a grammar system is a mathematical grammar. Mathematical grammar may serve as a mathematical basis for grammatical theory.

I suggest how mathematical grammar may accord with current theories of grammar, namely generative grammar, dependency grammar, and relational grammar, all of which purport to model universal grammar. Therefore, in general, mathematical grammar may serve as a mathematical basis for universal grammar, which is the study of principles of grammar universal to all natural languages; in other words, it is the study of the foundations of natural languages.

The general formal language, therefore, may be an approach and mathematical framework for both the foundations of mathematics and of language in general.

BIBLIOGRAPHY

[1] D. ALLERTON, *Valency Grammar*, vol. 13 of Encyclopedia of language and Linguistics, Elsevier, New York, 2006.

[2] H. B. ENDERTON, *A Mathematical Introduction to Logic*, Harcourt/Academic Press, Burlington, Massachusetts, 2001.

[3] G. EVANS, *Pronouns, quantifiers, and relative clauses*, Canadian Journal of Philosophy, (1977), pp. 46–115.

[4] J. FERREIROS, *The road to modern logic-an interpretation*, The Bulletin of Symbolic Logic, 7 (1984), pp. 441–484.

[5] L. FRAENKEL, BAR-HILLEL, *Foundations of Set Theory*, vol. 67 of Studies in Logic and the Foundations of Mathematics, Elsevier, London, 1973.

[6] R. FREIDIN, *Principles and Parameters Framework of Generative Grammar*, vol. 10 of Encyclopedia of language and Linguistics, Elsevier, New York, 2006.

[7] S. T. KUHN, *Is*, The Cambridge Dictionary of Philosophy.

[8] J. KUJIFF, *Dependency Grammar*, vol. 3 of Encyclopedia of language and Linguistics, Elsevier, New York, 2006.

[9] J. LUKASIEWICZ, *Aristotle's Syllogistic from the Standpoint of Modern Formal Logic*, Oxford University Press, Oxford, 1958.

[10] E. MENDELSON, *Introduction to Mathematical Logic*, Chapman and Hall/CRC, Boca Raton, Florida, 1997.

[11] J. MOORE, *Grammatical Relations and Arc-Pair Grammar*, vol. 4 of Encyclopedia of language and Linguistics, Elsevier, New York, 2006.

[12] F. ORILIA AND C. SWOYER, *Properties*, The Stanford Encyclopedia of Philosophy, http://plato.stanford.edu/archives/win2016/entries/properties/, Edward N. Zalta (ed.), Winter 2016 Edition.

[13] A. N. PRIOR, *Logic, History of*, vol. 4 of The Encyclopedia of Philosophy, Macmillan, New York, 1967.

[14] A. N. PRIOR, *Logic, Traditional*, vol. 5 of The Encyclopedia of Philosophy, Macmillan, New York, 1967.

[15] J. ROORYCE, *Generative Grammar*, vol. 4 of Encyclopedia of language and Linguistics, Elsevier, New York, 2006.

[16] F. SOMMERS, *The calculus of terms*, Oxford Journals, 79 (1970), pp. 1–39.

[17] F. SOMMERS, *The Logic of Natural Language*, Clarendon Press, Oxford, 1982.

[18] R. VAN VALIN JR, *Functional Relations*, vol. 4 of Encyclopedia of language and Linguistics, Elsevier, New York, 2006.