Alternative Technologies

Logic for Serious Database Folks Series

by David McGoveran, Alternative Technologies

LEGAL NOTICE AND LICENSE

IMPORTANT CONTENT NOTICE

It is anticipated that many readers of this document will have some familiarity with the writings of other authors, especially but not limited to E. F. Codd ("EFC"), C. J. Date ("CJD"), and H. Darwen ("HD"), or familiarity with Date and Darwen's The Third Manifesto ("TTM") and related writings. Material in this series will not comply with all of TTM and definitions therein, nor with any other interpretation of the work of EFC. Readers should not assume otherwise of an independent work. I will, from time to time, refer to specific passages in the works of EFC, CJD and HD for purposes of commentary or to illustrate some issue. When I disagree with a some construct by EFC, CJD, or HD and am aware of it, I will say so and give a justification.

CONTACT BY REVIEWERS

Reviewers are encouraged to convey their comments and criticisms to me directly via email addressed to **mcgoveran@AlternativeTech.com**., and including "REVIEW" in the subject line. Please provide document title, revision date, and page when referencing any passage. I will do my best to respond in a timely manner and will try to answer specific questions if there is a reasonably short answer.

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

Alternative Technologies

Chapter 5: How Formal Logical Systems Differ
Logic for Serious Database Folks Series

by David McGoveran, Alternative Technologies

"*Whatever Logic is good enough to tell me is worth writing down...*" – Lewis Carroll

I. INTRODUCTION

As you might have guessed by now, there are many ways in which the components of formal systems, in general, and of formal logical systems in particular, can vary. Any two formal logical systems that differ in how their components are defined will typically not be equivalent in terms of their power (defined formally below): either the deductions possible or the subjects that can interpret them will differ, possibly both.

How a formal logical system is defined (or constructed) determines its characteristic properties, called ***meta-mathematical properties***. A formal logical system, or some element within it, must be proven to have any particular meta-mathematical property. For simple properties, this is trivial and is largely a matter of inspection. For more sophisticated or subtle properties, a formal proof must be given. Such proofs are given in a metalanguage that is necessarily at least as *expressively powerful* (a term defined below) as the system's object language and quite possibly more powerful. Of special interest are those properties that can be proven using the system's object language as the metalangauge. We often say that such properties are then "provable within the system."

Meta-mathematical properties have many consequences. Relative to some other system, a formal logical system may be any or all of, for example:

(i) easier or more difficult to understand,

(ii) better or poorer for representing aspects of some subject, and,

www.AlternativeTech.com mcgoveran@AlternativeTech.com
DRAFT Chapter 05 – March 23, 2016
Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

(iii) better or poorer for formal, automated reasoning.

Even more important for implementing a computer software system (such as a DBMS and its query language) based on a formal logical system, the definition of the formal logical system chosen is crucial:

(iv) some formal logical systems are reliable (in some specific formal sense), while others are not, and,

(v) for some formal logical system, algorithms exist for testing certain properties while no such algorithm is possible in other systems.

The first three characteristics cited above, (*i*), (*ii*), and (*iii*), also have important consequences for the practice of designing a database intended to represent some system in the physical world (a kind of subject). As discussed in a previous article, representing a subject in terms of a formal logical system so that the system faithfully describes that subject requires care. One must identify which elements one wishes to consider the primitive elements of the subject and from which the rest of the representation will be constructed. The remaining elements of the subject, including relationships and complex entities, are then understood as being combinations of these primitive elements.

If the formal logical system is not powerful enough, it may not be able to represent all the primitive elements or their combinations. If too powerful, the system may "overstate" the subject, leading the user to deduce the existence of abstract constructions or relationships that have no correspondence in the subject.

In short, both situations are undesirable and exemplify "bad interpretation". Of course, when applied to the problem of database design, these situations imply that choices must be made in the processes of conceptual modeling and logical modeling to enable proper correspondences between the subject and the formal logical system. As we shall see, database design is a (typically iterative) process of, on the one hand (a) delimiting a subject to selected elements designated as either primitive or non-primitive, and determining how elements of the subject are inter-related; and (b) selecting a fragment of a formal logical system such that all and only the selected elements and relationships found in the subject are represented by that fragment and that the representation is faithful to the subject. The former is a matter of application scope, while the latter is a matter of identifying an appropriate formal logical system. Altogether, this is a process of making trades-off intelligently. If the formal logical system is insufficiently powerful for the purpose, then either another, more powerful, system must be selected (or defined) or the original subject must be delimited differently possibly excluding certain elements or relationships as being *out of scope*.

This sounds like a straightforward, if complex, process. Unfortunately, there exists a "gotcha." Certain properties of formal logical systems determine the characteristics (iv) and (v) cited above

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

in unexpected ways. It turns out that, in general, the more powerful the formal logical system, the less likely that the system will be reliable (in certain precise senses to be defined more formally below) and the less likely for there to exist algorithms that can identify certain desirable properties (again, to be more formally defined below). Thus, the choice of a formal logical system (whether explicitly or implicitly as a consequence of other choices) is of paramount importance. To understand this issue, we have to understand the properties that differentiate formal logical systems, what affects them, and at least some of their consequences.

Interestingly, one of the most powerful formal logical systems is that which underlies most (if not all) so-called general purpose programming languages. Such languages are *Turing complete* (TC), also known as *computationally complete*. For the moment and very simplistically, we can think of a TC language as one that can express any procedure implementable on a computer. It is because their expressive power is so great that software developers must test their programs for logical correctness[1] and still miss "bugs": there can be no algorithm that will test any conceivable program written in a TC language for logical correctness.

It is tempting to simply blame the programmer for making errors and, if humans were infallible, that might be reasonable. However, if a general purpose algorithm for finding errors were possible, then the programmer could that use that a program that implemented that algorithm to identify errors in whatever code was written, and the programmer could then fix them. Of course, custom test programs and "test suites" can be written to check any specific program, and tools exist for making this as easy as possible. In my experience few programmers understand that the source of the need to laboriously design, develop, and test software for logical correctness lies in the power of the logical system underpinning TC languages. In a sense, the extraordinary power of TC languages is the cause of every software bug you've ever seen or heard about.

In this article, we will identify the properties that can differ among of formal logical systems in an organized fashion and discuss some of the consequences of those differences. In later articles, we will use this knowledge to make informed choices in database theory and practice.[2]

Our task in this article is non-trivial for several reasons. First, the terminology for properties of formal logical systems as found in the literature is not universal. The meaning of a term may have changed over time, different authors may use the same term to mean different concepts, terms may be used in a specialized way in the context of some particular system, and so on. More difficult to sort out, sometimes renowned logicians simply disagree.

Ideally, we could provide a kind of standard glossary of terms that would allow the reader to

---

[1] Not only must a program meet perform the desired functions as given by functional requirements (however collected), it must do so in a logically correct manner.

[2] These considerations are applicable to any effort involving knowledge discovery, representation, and reasoning in general, database theory and practice being just one exemplar.

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

read and understand the literature unambiguously. Sadly that is not possible. In what follows, I have attempted to rationalize the lexicon using several complementary strategies. Where possible, I have attempted to provide generalized definitions with guidance to the reader as to how to specialize that definition when applying the term to a particular logical system. Where unqualified terms are used ambiguously in the literature, I have attempted to remove the ambiguity by qualifying the term before providing a specific definition. Finally, I have provide a framework for categorizing properties.

Broadly and somewhat informally, *metamathematical properties* may be grouped into two categories. The first category consists of properties that pertain exclusively to the Deduction Subsystem and are *proof theoretic*. We will further divide these into *structural properties* and *syntactic properties*. Structural properties pertain to individual *components* of the Deduction Subsystem including, for example, vocabulary, axioms, and inferences rules. Structural properties are definitional. By contrast, syntactic properties are consequences of those structural properties, possibly in combination and in terms of how they may be used. Syntactic properties must be shown by meta-theoretic proof to apply to the formal logical system.

The second category consists of properties that pertain exclusively to the Interpretation Subsystem and are *model theoretic*. We will further divide these into *structural properties* and *semantic properties*. Model theoretic structural properties pertain to individual *components* of the Interpretation Subsystem including, for example, differences in the number and type of truth values, and in the evaluation procedure. Structural properties are, again, definitional. By contrast, semantic properties pertain to interpretations of the formal deductive language. Semantic properties must be shown by meta-theoretic proof to apply to the formal logical system.

For certain model theoretic properties, a formal logical system is said to have the property if and only if it meets the definition of the property for *all permissible interpretations*. Since the semantics implied by any particular interpretation can have no effect, such properties are sometimes considered to be *syntactic* rather than semantic. However, such properties assume the existence of interpretations and are impossible to define without them: the system must be *interpretable*. For this reason we will include them in the category of *semantic* properties. Certain other model theoretic properties pertain to special *relationships* that arise between the Deduction Subsystem and the Interpretation Subsystem. As we will see, properties in this latter group are among the most important because they tell us what, and what not, to expect of the formal system.

I have chosen the foregoing categorization both for pedagogical reasons and in order to emphasize the importance of distinguishing proof theory on the one hand from model theory on the other. Proofs are constructed according to the axioms and rules of inference of the logical system. Proof theoretic properties may be said to be purely syntactic in the sense that they do not rely upon any specific interpretation, on the existence of some interpretation, or even on all permissible interpretations. Interpretations and evaluations are constructed according to meaning

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

assignments, truth assignments, and the evaluation procedure. Every model theoretic property relies upon the nature of interpretations in some manner: either every permissible interpretation will induce or have some defining characteristic of the property, or at least one interpretation exists that induces or has the defining characteristic of the property.

Other ways to categorize properties are sometimes used. For example, some authors differentiate ***system properties*** (properties that pertain to the system as a whole) from ***wff properties*** (properties that pertain to a particular wff or set of wffs). In some cases, the definition of a system property requires that a set of wffs (sometimes all wffs) have some wff property or properties. We will point these distinctions out from time to time.

This article can serve as a reference for the hard work to come when we characterize and compare certain specific formal logical systems such as propositional logic, first order predicate logic, second order predicate logic, lambda calculus (and other computationally complete systems), and many-valued logics. I suggest reading this article through a few times to gain familiarity with the described properties' definitions, their interrelationships, and their categorization. Then, as you read subsequent articles in this series, I suggest keeping this article handy as a reference.

Some General Concepts

When characterizing formal logical systems, we shall often find it helpful to characterize the sets of a system's components, especially the size (in particular, the cardinality). Before we dive in[3], the following terms will be helpful (either in the present article or a future article):

- *countable* – A set is countable if its members can be put in a one-to-one correspondence with a subset of the natural numbers {1, …, **n**}.

- *finite* – A set is finite if its members are countable, with the penultimate correspondence **n** being to a specific natural number when counted.

- *infinite* – A set is infinite if its members are not finite (*i*) being not countable or (*ii*) being countable but there being no "penultimate correspondence".

- *denumerable* a.k.a. *countably infinite* – A set is denumerable (countably infinite) if its members are countable, but not finite (there being no "penultimate correspondence"). For example, the set of integers is denumerable.

[3] For further discussion of set theory see the second article in this series "Set Theory and Meta-Language".

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

- *non-denumerable* – An infinite set is non-denumerable if its members cannot be put into one-to-one correspondence with the natural numbers. For example, the set of real numbers is non-denumerable.

Certain properties of sets or their members apply to numerous structures relevant to formal logical systems.

- *bounded* – A numeric characteristic of a set, possibly part of its specification, is bounded if it is has either a least (minimum) or a greatest (maximum) value. For example, a set such as "the real numbers greater than 10" is infinite but bounded from below because it has a minimum member.

- *sequence* – A sequence is an ordering of objects called the terms of the sequence, the same object possibly appearing as multiple terms. A sequence of **n** terms is called an **n**-tuple.

- *enumeration* – An enumeration of a set ***S*** is a finite or denumerable sequence **Sn** in which every member of ***S*** is a term and every term of **Sn** is a member of ***S***. The members of ***S*** may appear in **Sn** multiple times.

- *enumerable* – A set ***S*** is enumerable if there provably exists at least one enumeration **Sn** of ***S***.

- *effective procedure* – A procedure (a.k.a., a method) is effective if, when followed correctly and for as many steps as necessary, provably produces a correct answer after completion of a finite number of steps. The precise number of steps need not be predictable.

- *effective enumeration* – An enumeration **Sn** is an effective enumeration if there provably exists an effective method for finding the $\mathbf{n}^{th}$ term of the **Sn**. The method need not be known.

- *primitive element* – An element is described as ***primitive*** if and only if it has no identified components and is not subject to analysis (whether by fact or fiat). For any set ***S*** of elements and a set of prescribed methods ***M*** for combining or modifying elements into new elements, a subset ***S1*** of ***S*** may be designated as ***primitive***, meaning those elements are *a priori* not a combination of any other elements. Notice that what constitutes a primitive set is a matter of choice and not an inherent property: any one subset, but only one, may be designated as primitive.

- *derived element* – An element is described as ***derived*** if and only if it is the result of a method, either directly or indirectly, applied to primitive elements. Typically the concept of being derived will be qualified by the permissible methods by which such elements may be obtained.

Notice that, for any non-empty set ***S*** of elements and a set of prescribed methods ***M*** on those

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

elements, ***S*** may be partitioned into two disjoint subjects: one whose members are designated as *primitive elements* and one whose members are *derived* - derivable from *primitive elements* by the set of methods ***M***. The set of derived elements may be empty, but the set of primitive elements cannot.

- *independent* – A subset ***S1*** of a set ***S*** of elements is characterized as ***independent*** if it is impossible for any particular element in the subset ***S1*** to be expressed or defined as the result of a some combination of methods in ***M*** on the other elements of ***S***. Typically the concept of independence will be qualified by the means by which such combinations may be obtained. For any subset ***S1*** of ***S*** and any method (or methods) of combining elements into new elements, ***S1*** is said to be an ***independent*** set if and only if each of those elements cannot be a result of some combination of prescribed methods on any other elements in ***S***.

- *minimal* – A set of elements is ***minimal*** with respect to a property (e.g., being composed of primitive elements) if and only if no other set of the type has smaller cardinality and still has the property of independence.

Unqualified (vs. Qualified) Terms

Many terms used in the study of logic and metalogic, and those referring to the properties of formal logical systems, are ambiguous when used without qualification. In many cases, there exist both a syntactic concept and a semantic concept that share the same unqualified name. In some cases, there are multiple unrelated properties that share the same unqualified name, perhaps as a result of historical accident as the study of logic evolved. As a result, it is crucial to use qualified names for properties or, if qualifications are not used, that the context makes it clear what that qualification would be. Unfortunately, the assumed context can be obscure in the literature and so the reader must be extremely cautious (especially when quoting or relying on some authority).

Often, there is no uniformly accepted method of determining whether or not a system or a wff has some general property, possibly because there is disagreement as to how the general concept should be made precise. It has become common practice to define only those variants of the general property that can be made precise, and to refer to these variants with qualified names.

One of the most fundamental concepts of logic is the *validity* of a wff. Most readers will have some intuitive sense, not necessarily of validity, but of *invalidity* – a sense that "something is not right." Unfortunately, "valid" is often erroneously thought to mean "true." The distinction between the concepts of "validity" and "truth" is essential to an understanding of formal logical systems. Be that as it may, there is no uniformly acceptable method of determining whether a wff is valid in a general sense. Instead, we define validity in more narrow senses, including *syntactic validity* and *semantic validity* (a.k.a., *logical validity*).

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

Most of this article is devoted to defining qualified terms for properties. However, it will be helpful for the reader to be aware of, and have a sense of, certain important but ambiguous, unqualified terms. Among the important unqualified property terms (in bold and italics) are:

- *power* – By ***a system's power***, we mean a measure of an intrinsic capability of the system. A system's power is often given relative to that of some system of known capability. System power can refer to either the system's intrinsic deductive power or its intrinsic expressive power.

- *equivalence* – By an ***equivalence property***, we mean a measure shared by two systems, inference rules, operators, or wffs so that the two may be judged equivalent with respect to that measure. The measure may identify deductive, expressive, truth functional, or semantic capabilities.

- *validity* – A wff ***f*** (or inference ending in ***f***) is ***valid*** with respect to some property if the wff ***f*** obtains in some specified sense when that property holds. A valid wff or inference can be valid with respect to provability (a.k.a., *syntactic validity*) or valid with respect to satisfiability (a.k.a., *semantic validity* or *logical validity*).

- *entailment* and *consequence* – A set ***S*** of wffs ***entails*** a wff ***f*** (alternatively, the wff is a ***consequent*** of a set of wffs) if the wff ***f*** follows from that set ***S*** of wffs in some specified way. Consequence can be either *logical consequence* or *semantic consequence*.

- *soundness* – By a ***soundness property***, we mean that a system or a set of inference rules preserves some specified property. The system or set of inference rules is then said to be ***sound with respect to the specified property***. Otherwise the system or set of inference rules is said to be ***unsound*** *(with respect to the property)*. Typically, soundness refers to soundness with respect to *semantic validity*. Notice that a wff is sound with respect to provability if it is provable, and a set of inference rules is sound with respect to provability by definition.

- *consistency* – By a ***consistency property*** we mean that <u>at most</u> certain wffs (i.e., those that have a particular property – such as of *validity* – see below) are provable. In other words, the particular property is a necessary, but not sufficient, condition for provability. A formal logical system is then said to be ***consistent with respect to the particular property***. Otherwise the system is said to be ***inconsistent*** *(with respect to the property)*. Consistency can be proof theoretic (a.k.a., *p-consistent* or *negation consistent*), relative, absolute, relative, or model theoretic.

- *completeness* – By a ***completeness property***, we mean that <u>at least some</u> wffs (i.e., all those that have a particular property – such as *validity* – see below) are provable and therefore are theorems of the system. In other words, the particular property is a sufficient condition for provability. A formal logical system is then said to be ***complete with respect to the***

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

particular property. Otherwise the system is said to be ***incomplete*** *(with respect to the property)*. Completeness can be expressive, truth functional, deductive (a.k.a. with respect to provability, syntactic, or maximally), strongly, negation, refutation, or semantic.

- *decidability* – By a ***decidability property***, we mean that it is provable that an effective procedure exists that can determine (i.e., decide) whether any specific wff of the system either has or does not have the specified property. A formal logical system is then said to be ***decidable with respect to the particular property***. Otherwise the system is said to be ***undecidable*** *(with respect to the property)*. Decidability can be with respect to provability, satisfiability, truth valuation, or validity.

Unfortunately, all the foregoing terms, as well as their antonyms, are sometimes used without qualification to refer to a number of properties in the literature, each having a distinct formal definition. The literature often fails to explicitly differentiate these properties, relying on context and the reader's expertise to determine which property is being discussed. To make matters more confusing, under certain circumstances two otherwise distinct properties may each imply the other in some formal logical systems and authors then use the terms interchangeably.

As we will see, the qualified versions of these properties are of great importance to our subject matter: whether a formal logical system (or wff, rule of inference, and the like) possesses these properties determines whether they are or are not suitable as a foundation for a DBMS or, least, determines the advantages and disadvantages of using them for that purpose. We define the qualified properties below, dividing them into proof theoretic properties and model-theoretic properties.

Properties have consequences, and where reasonable, these are described with each property in the sections that follow.

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

II. PROOF-THEORETIC STRUCTURAL PROPERTIES

Proof-theoretic structural properties can be grouped into properties pertaining to vocabulary, formation rules, axioms, inference rules. We look at each of these groups below, making the ways systems can differ explicit and, where reasonable to do so, pointing out the potential impact of such differences.

The Vocabulary

The vocabulary of formal system can differ in terms of its logical symbols, its non-logical symbols, or both.

Logical Symbols

The logical symbols of a vocabulary are typically divided into grouping indicators and operator symbols (*the operator symbol set*). A certain number of the logical symbols for operators may be designated as primitive (*the primitive operator symbol set*). All non-primitive operators symbols are then necessarily defined in terms of combinations of primitive operator symbols (according to either the formation rules or the inference rules), and then denoted via a unique logical symbol. Recall that "primitive" is a designation, not an intrinsic property to be discovered.

Logical Symbol Set Cardinality – The number of logical symbols may be finite (some specific number) or infinite.[4] For the sake of generality, the number of logical symbols is often taken to be infinite as part of the formal definition and then reduced to a finite number in practice.

Operator Symbol Set Cardinality – The cardinality of the operator set may be finite or infinite so long as this is consistent with the cardinality of the logical symbol set. Having a large number of operators, let alone an infinite set, negatively affects comprehension and usability.

Primitive Operator Symbol Set Cardinality –The number of primitive operator symbols may be finite or infinite (although the latter is rare).[5] The cardinality of the set may also be given as bounded, countable, or uncountable. The number of primitive operator symbols affects both comprehension and usability. If too small, expressions involving those operator symbols become more verbose and difficult to understand. If too large, there are likely to be multiple, equivalent expressions.

Primitive Operator Symbol Set Independence – We say that a primitive operator symbol set is *independent* if it is impossible for any primitive operator symbols in the set to be expressed in terms of the others. A primitive operator symbol set is either independent or not. We say that two

[4] The set may be of transfinite cardinality, although this possibility will not concern us in this work.
[5] The set may be of transfinite cardinality but, again, this will not concern us herein.

sets of operator symbols are *equivalent* if every operator symbol in one can be defined in terms of a composition of one or more operator symbols in the second and vice-versa. For any non-independent set of operator symbols, there exists a set of operator symbols that is both independent and equivalent to the original set. If the primitive operator symbol set is not independent, two or more wffs can express the same thing while appearing completely different.

Minimal Primitive Operator Symbol Set – The number of operator symbols may or may not be minimal in the sense that, of all possible primitive operator symbol sets, any other primitive operator symbol set is either non-independent or has greater cardinality.

Canonical Interpretation – Each primitive operator symbol will have a canonical interpretation or intended meaning (as given by the definer of the formal system). The primitive operator symbol set thus corresponds to a set of operations as understood in the canonical interpretation.

For two formal systems, any of the foregoing may differ.

Non-Logical Symbols

The set of non-logical symbols, and the subsets discussed below, may have either *finite* or *infinite* cardinality. They may also be transfinite, although this possibility will not concern us in this work. For the sake of generality, the number of non-logical symbols is often taken to be infinite (at least in principle). A certain finite number of non-logical symbols may be designated as primitive. Any combination of primitive non-logical symbols may then be given a non-logical symbol (intended to designate a more complex entity). This approach permits a potentially large number of (abstract) entities to be given a symbolic representation.

In a formal logical system, certain non-logical symbols are designated as constants, among them constants interpreted as *truth indicators* under the canonical interpretation. In keeping with the distinction between syntax and semantics as presented herein, these truth indicators belong to the Deduction Subsystem and so are considered distinct from *truth values*, which properly belong to the Interpretation Subsystem.[6]

Non-Logical Symbol Set Cardinality – The number of non-logical symbols may be finite (some specific number) or infinite. The cardinality of the set may also be given as bounded, countable, or uncountable.

Non-Logical Symbol Set Independence – The primitive non-logical symbol set may or may not be *independent* in the sense that it is impossible for any particular non-logical symbol to be expressed in terms of the others.

[6] In a logical calculus as defined, for example, by Carnap (The Logical Syntax of Language, 2010) truth indicators and truth values are not distinguished.

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

Non-Logical Symbol Independent Set Cardinality – The number of independent non-logical symbols may be finite or infinite (although the latter is rare). The cardinality of the set may also be given as bounded, countable, or uncountable.

Minimal Primitive Non-Logical Symbol Set – The number of non-logical symbols may or may not be minimal in the sense that, of all possible primitive non-logical symbols sets, any other primitive non-logical symbols set is either non-independent or has greater cardinality.

Non-Logical Symbol Set Categories– Non-logical symbols may be partitioned into a number of primitive, independent categories[7] (each a *primitive category set*). From these *primitive category sets*, derived *category sets* may be defined. The *primitive category set* might or might not be minimal in the sense that, of all possible primitive category sets, any other primitive category set is either non-independent or has greater cardinality.

Truth Indicator Set Cardinality – The number of truth indicators may be finite or infinite. The latter is rare and, with few exceptions, requires an unusually abstract canonical interpretation. The cardinality of the set may also be given as bounded, countable, or uncountable.

The Formation Rules

Every formal logical system is defined as having one of many different possible sets of formation rules. Each particular formation rule set may be partially characterized in terms of its cardinality (the number of rules) and the specific rules. The formation rules determine what constitutes a wff and so affect any wff property and any system property that is required to apply to some set of wffs (possibly all wffs).

Formation rules specify a grammar using what a *syntax language*, distinct from the object language. There are many syntax languages[8] that can be used to specify ***equivalent formation grammars*** – grammars that describe the same set of wffs. Proving that two formation rule sets do or do not specify the same set of wffs when applied to the same vocabulary can be a challenge. Although the following properties are helpful in comparing formal logical systems, two formation rule sets with the same properties are not necessarily equivalent.

Formation Rule Set Cardinality – The number of distinct rules may be either finite or infinite, and if infinite may be countably infinite or not. Typically, one strives to define systems using a relatively small number of formation rules as this aids both comprehension and usability.

---

[7] Although a primitive set of categories might not be independent, but if not then such a set may be defined from it.
[8] Preferred syntax languages often have recursive grammars or regular grammars. We will define these in a later article.

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

Formation Rule Set Independence – The formation rule set may or may not be *independent* in the sense that it is impossible for any particular formation rule to be expressed in terms of the others.

Minimal Formation Rule Set – If no formation rule can be eliminated from the formation rule set without reducing the set of wffs described, we say the formation rule set is *minimal*.

Recursive Formation Rule Set – A formation rule set is said to be a *recursive* if they provides a recursive definition of a wff. One or more wffs having a particular form are chosen as "base" wffs, thereby providing the base of the recursion. Then, rules are given for creating more complex wffs by combining wffs and logical symbols, thereby giving the recursion or induction. For example, if any individual non-logical symbols such as P, Q, and R designate wffs and ∧ is a logical symbol designating an operator, then P ∧ Q is also a wff. By recursion it follows that P ∧ Q ∧ R is also a wff. Notice that recursiveness is a purely syntactic concept – no specific meanings have been given to the symbols involved.

The Axioms

Formal systems may or may not be axiomatic. An axiomatic formal system has an explicitly given set of axioms. As previously noted, each axiom is a wff that is *a priori* semantically valid (see the definition of *semantic validity* below).

Every formal logical system is defined as having one of many different possible sets of axioms (*axiom sets*). Each particular *axiom sets* may be partially characterized in terms of its cardinality (the number of axioms, possibly zero) and the specific axioms. Because axioms are necessarily the initial wffs of every proof sequence, they affect any property that depends upon what can and cannot be proved.

Axiom Set Cardinality – The number of distinct axioms may be either finite (possibly zero) or infinite, and if infinite may be countably infinite or not. When the number of axioms is large of possibly infinite, we may instead specify a set of *axiom schemata* – that is, a set of axiom forms such that any wff having the specified form is an axiom. A large number of axioms affects comprehension and usability of the Deduction Subsystem, and increases the likelihood of an inconsistency among them. We will give an example of axiom schemata in a later article when we address a specific formal logical system.

Axiom Set Independence – An axiom set is said to be *axiomatically independent* (a.k.a., *deductively independent*) *with respect to an inference rule set* if it is impossible for any particular axiom to be derived from the others, given the set of inference rules. The rules in the set are thus mutually independent. It is typically possible to find multiple independent axiom sets each of which are equivalent in the sense that they can be used to deduce the same set of proof

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

sequences using some specific inference rule set. If a particular wff in an axiom set is independent of the others then either it or its negation may be assumed without introducing an inconsistency. If the axiom set is axiomatically independent, it is *a priori* consistent.

Minimal Axiom Set – An axiom set is *minimal with respect to an inference rule set* if no axiom can be eliminated from the axiom set without reducing the set of wffs that can be deduced. Note that there may exist multiple distinct, but minimal axiom sets. As the axiom set decreases in size, proof sequences tend to become longer and vice-versa. However, the ability to create proof sequences and understand individual inference steps can be facilitated.

The Inference Rules

Every formal logical system is defined as having one of many different possible sets of inference rules. Each particular inference rule set may be partially characterized in terms of its cardinality (the number of rules) and the specific rules. Because the inference rules determine what constitutes a proof sequence, they – like axioms – affect any property that depends upon what can and cannot be proved.

Inference rules, like formation rules, are specified in a *syntax language* (again, distinct from the object language) and provide an ***inferential grammar*** for the formal logical system. There are many ways to specify ***inferential grammars***, some of which are equivalent**.** Two formal logical systems are said to have ***equivalent inferential grammars*** if their inference rules can be used to construct or deduce the same sets of proof sequences from the same set of axioms. Proving that two inference rule sets do or do not specify the same set of proof sequences when applied to the same axiom set can be a challenge. The following properties are helpful in comparing formal logical systems. However, it is important to remember that, even if these properties are the same, two inference rule sets do not necessarily provide equivalent inferential grammars.

Inference Rule Set Cardinality – The number of distinct inference rules may be either finite (possibly zero) or infinite, and if infinite may be countably infinite or not. When the number of inference rules is infinite, we may instead specify a set of *inference schemata* – that is, a set of inference rule forms such that any inference step having the specified form is an application of the inference schemata. We will give an example of inference schemata in a later article in a later article when we address a specific formal logical system.

Inference Rule Set Independence – An inference rule set is *independent* if it is impossible to infer any particular inference rule from the combination of the others. They are thus mutually independent. It is typically possible to find multiple independent inference rule sets each of which are equivalent in the sense that they can be used to deduce the same set of proof sequences from some specific axiom set.

Minimal Inference Rule Set – An inference rule set is *minimal* if no inference rule can be eliminated from the inference set without reducing the set of wffs that can be deduced. We will

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

say that the inference rule set is *globally minimal* if, in addition to being minimal, there exists no inference rule set of smaller cardinality. Note that there may exist multiple distinct, but minimal inference rule sets.

The Trades-off Between Axioms and Inference Rules

It is well known that the number of inference rules can be reduced by increasing the number of axioms or replacing axioms with *axiom schemata*. This is especially useful if the number of inference rules is zero. Conversely, sometimes the number of axioms can be reduced by increasing the number of inference rules, or by introducing *inference schemata*. Again, this is especially useful if the number of axioms is zero

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

III. PROOF-THEORETIC SYNTACTIC PROPERTIES

Proof theory uses the rules of inference to study whether or not a wff can be deduced and proven, whether or not a set of wffs are *deductively equivalent*, and whether or not the formal logical system has certain properties of *consistency* and *decidability*.

The definition of a formal logical system is meant to specify a set of symbols, operations, and rules for their manipulation that can be applied in a purely mechanical fashion: that is, the process of determining whether an expression belongs to the vocabulary, is a wff, or is an axiom, must not be based on judgment or on the outcome of some random event.

We do not mean that the system's rules are necessarily intended to be followed by a machine rather than a person. However, loosely speaking, any application of the rules can be performed by a machine as well as a person, and will yield the same result each time it is repeated. Determining whether or not a rule of inference has been properly applied must also be purely mechanical. As we will see, having rules specified in such a way that they can be mechanically applied does not necessarily mean that an algorithm exists for determining which rules to apply when[9] and, in general, such an algorithm will not exist.

Each possible combination of choices for the components that make up a formal logical system results in a distinct formal logical system. We can sometimes show two formal logical systems are equivalent in some sense. Usually these are trivial cases involving, for example, inconsequential differences of representation such as using the Greek alphabet in place of the English alphabet for symbols, or if we change which wffs are axioms and so which must be theorems.

Deductive Power and Deductive Equivalence

Formal logical systems rarely have the same ***deductive power***, defined as the set of theorems that can be derived. If two systems have equal deductive power, they may be said to have the syntactic property of ***deductive equivalence*** (***of systems***) and so are ***deductively equivalent.*** Sometimes one formal logical system $\mathcal{L}_1$ is a deductively equivalent to a subset of a second formal logical system $\mathcal{L}_2$, in which case any wff provable in $\mathcal{L}_2$ is provable in $\mathcal{L}_1$ and we say that $\mathcal{L}_1$ is deductively more powerful than $\mathcal{L}_2$. ***Deductive equivalence (of wffs)*** applies the property to two wffs: Two wffs belonging to the same formal logical system are said to be deductively equivalent (a.k.a., syntactically equivalent) if one can be deduced from the other and vice-versa.

As a matter of applied logic, the selection of a formal logical system for some particular purpose will often be dictated to a significant degree by the requirement of sufficient deductive power.

---

[9] We are, in effect, excluding the class of programs called *theorem provers* which use various heuristics and machine learning methods, and are non-deterministic.

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

Perhaps counter-intuitively, however, we will often need to avoid selecting a system with too much deductive power. The semantic counterpart to deductive power – *expressive power* – is also crucial and will be discussed in the model-theoretic section on semantic properties below.

Logical Entailment

A set of wffs ***S*** ***logically entails*** a wff ***f*** (written "$\boldsymbol{S \vdash f}$") if and only if there is a formal proof of ***f*** (the consequent) from ***S*** (the premises); that is, if and only if ***f*** is ***provable*** from ***S***. Equivalently, we say that ***f*** is a ***logical consequence*** of ***S***. *Logical entailment* is also known as *logical consequence, syntactic consequence,* or *logical implication*. We will see below that logical entailment is a important prerequisite for a number of other properties. Logical entailment and logical consequence are not to be confused with their model-theoretic counterparts: *semantic entailment* and *semantic consequence*, both defined in the model theoretic section below.

Syntactic Validity

We say that an inference is a ***syntactically valid inference*** (a.k.a., *valid with respect to provability*) if and only if the concluding wff premises can be derived from the (usually immediate) premises via a rule(s) of inference. The conclusion of a syntactically valid inference is sometimes referred to as a ***syntactically valid wff***. Recall that we defined a ***formal proof*** of a wff ***f*** as a sequence of wffs (including axioms) ending in ***f***, each the result of applying the rules of inference to the preceding wffs. (The initial wffs in the sequence are typically axioms.) Therefore, a formal proof is a ***syntactically valid sequence*** of inferences. A wff ***f*** is said to be ***provable*** if there exists a proof of ***f***. Every theorem is, by definition, a syntactically valid wff.

(Syntactic) Consistency

A formal logical system is said to be ***proof-theoretically consistent*** (a.k.a., *p-consistent* or *negation consistent*) if no two provable wffs contradict each other (i.e., it is not the case that there exists a wff **A** for which both **A** and ¬**A** are theorems).

A system is ***absolutely consistent*** if and only if at least one wff exists that is not a theorem of the system. Note that if this is not the case, then every wff and its negation are theorems.

A system $\mathcal{L}_1$ that is proven consistent under the assumption that a different system $\mathcal{L}_2$ is consistent is said to be ***relatively consistent*** or "***consistent relative to*** $\mathcal{L}_2$". The reader might better understand relatively consistency by thinking of it as "contingent consistency": if $\mathcal{L}_1$ ***is consistent relative to*** $\mathcal{L}_2$ then the consistency of $\mathcal{L}_1$ is contingent upon the consistency of $\mathcal{L}_2$. There may already exist a direct proof of the consistency of $\mathcal{L}_2$., and a proof of the consistency of $\mathcal{L}_1$ relative to $\mathcal{L}_2$ may well be easier to establish than a direct proof of the consistency of $\mathcal{L}_1$. Be aware, though, that $\mathcal{L}_2$ might in turn be relatively consistent.

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

Negation Completeness

A formal logical system is ***negation complete*** if for each truth valued wff ***f***, either ***f*** or ¬***f*** is a theorem. Negation completeness is the form of completeness that Gödel's Incompleteness Theorems address (we will discuss these theorems in more detail in a later article). A system is ***negation incomplete*** if, for at least one truth valued wff ***f*** such that (i) neither ***f*** nor ¬***f*** is a theorem, or (ii) ***f*** is *undecidable* (see below). In systems having certain additional properties, deductive completeness (a semantic property defined in the next section) implies negation completeness and negation completeness implies deductive completeness.

Deductive Decidability

A formal logical system is said to be ***decidable with respect to provability*** (a.k.a., ***p-decidable*** or ***deductively decidable***) if it is provable (via a meta-theorem) that there exists an algorithm – an effective, mechanical procedure – for determining if an arbitrary wff is provable or not. Alternatively, we may say that a procedure exists that decides whether every wff is or is not a theorem of the system. Such a procedure is said to *solve* the ***decision problem for provability***. Note that this does not necessarily mean that an effective procedure is known for finding a specific proof, only that one exists.

When it is proven that no algorithm exists to solve the decision problem for provability, the formal logical system is said to be ***p-undecidable***, ***deductively undecidable***, or ***undecidable for provability***. For a formal logical system that is p-undecidable theory, there exists wffs which are provable (or unprovable) but for which there is no algorithm that can make that determination. For p-undecidable systems then, we are in the unfortunate position of having to find an actual proof (of disproof) for potentially very complex wffs before we can assert they are (or are not) a theorem. Note that failing to find a proof does not enable us to conclude that the wff is not a theorem.

Most often, the unqualified terms "*decidable"* or "*decidability"* in the literature will refer to *decidability for provability*. However, the reader should read with caution as it is not always clear as to which type of decidability is intended when the terms "decidable" and "decidability" are used. We may also speak of a particular wff or a certain class of wffs of a system being decidable. While many systems are not decidable, it is often the case that a particular class of wffs belonging to the system – wffs that are of a particular form, or satisfying certain conditions regarding their form – are *decidable*. This can be an important property when a particular interpretation of the system can be shown to involve only wffs of the decidable class, especially if that interpretation is to be implemented on a computer. On the other hand, if the computer application is not limited to a decidable class, then certain capabilities cannot be automated (i.e., by definition no effective procedure exists).

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

IV. MODEL-THEORETIC STRUCTURAL PROPERTIES

It is sometimes said that model theory does not require any particular interpretation, since it deals with all possible truth value assignments. However, it is more accurate to say that it considers all possible interpretations, identifying those that are permissible and, in so doing, all possible truth value assignments. Throughout this section and the next, we consider properties that arise in consequence of the possible interpretations of a formal logical system (the subject of model theory).

Evaluation Language: The Truth Value Set

The Truth Value Set may be characterized by the number truth values and the type of each truth value. Of all the truth values, a subset will be designated as truth-like (or simply "*designated*") and a certain number will be designated as false-like (or simply "*anti-designated"*). Again, we remind the reader that the symbols "**/*𝓣*/**" and "**/*𝓕*/**" are used herein to signify arbitrary designated and anti-designated truth values.

For each truth value in the Truth Value Set, there exists a non-logical symbol that behaves as a constant. Every evaluation language requires a method of comparing truth values, which we will refer to here as a *truth value comparison relation*. Under the truth comparison relation, the members of the set of truth values may be *totally ordered*, *partially ordered*, or *unordered*. When the number of truth values is greater than two, the truth value comparison relation is usually given explicitly, although sometimes given implicitly by defining the Truth Value Set with an ordering.

For example, in a classical system, there are two truth values in the evaluation language, one of which is designated and the other of which is anti-designated. These truth values, which we symbolize as "**/T/**" and "**/F/**", usually correspond to "truth" and "falsity", respectively, in the subject. Hence, because there is exactly one designated truth value and one anti-designated truth value, the symbol "**/*𝓣*/**" is replaced by "**/T/**"and the symbol "**/*𝓕*/**" is replaced by "**/F/**" in all relevant definitions and discussions. Corresponding to these two truth values, there are two non-logical constants in the object language, which we symbolize throughout this series as "TRUE" and "FALSE". The truth value comparison relation is totally ordered.

Truth Value Set Cardinality – The number of truth values may be finite, infinite, or transfinite. The cardinality of the set may also be given as bounded, countable, or uncountable.

Designated Truth Value Set Cardinality – The number of designated truth values must be consistent with the Truth Value Set Cardinality, but otherwise may be finite, infinite, or transfinite. The cardinality of the set may also be given as bounded, countable, or uncountable.

Anti-designated Truth Value Set Cardinality – The number of anti-designated truth values must be consistent with the Truth Value Set Cardinality, but otherwise may be finite, infinite, or

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

transfinite. The cardinality of the set may also be given as bounded, countable, or uncountable. *Note Bene*: The number of anti-designated truth values is usually equal to, but can be less than, the difference between the number of truth values and the number of designated truth values.

Truth Value Comparison Relation – The truth value comparison relation may induce a total ordering, partial ordering, or no ordering on the Truth Value Set.

The Evaluation Procedure

As described earlier, the evaluation procedure is a method by which the truth value of a wff is computed under a particular assignment of truth values to each occurrence of the non-logical constants called truth indicators in the wff. Under the evaluation procedure:

- every truth indicator is replaced by its corresponding truth value,
- every logical operator symbol is replaced by a corresponding truth-valued transformation (this might not be a function),
- each non-logical symbol in a wff is assigned a meaning,
- every truth-valued term (e.g., constituent wffs) in the wff is assigned a truth value, and,
- a truth value for the wff is computed consistent with any grouping indicators.

If the truth value of every wff in a formal system can be computed from the truth values of the constituent non-logical symbols, the evaluation procedure is truth functional and the formal system is said to have ***truth functional semantics*** (or simply, to be ***truth functional***). If a system is truth functional, any semantic relationship among operands over and above their truth value assignments need not be considered in determining the truth value of a wff consisting of an operator having those operands. Of course, with respect to any specific interpretation, truth value assignments may themselves depend on the meaning assignments.

Semantic operators may be either *truth functional* or "*non-truth functional*." By non-truth functional we mean that they cannot be given a truth functional semantics. This distinction is important because it is possible to give a non-truth functional semantics for most (perhaps all) truth functional operators, and many formal logical systems that are presented using non-truth functional semantics can, in fact, be given a truth functional semantics at the cost of greater complexity. The "and then" connective of English is an explicit, simple example of a non-truth functional operator: the truth value of sentences using it cannot be determined solely from the truth values assigned to the conjuncts. Other non-truth functional operators include certain, though not all, operators defined in systems of many-valued logics, modal logics, probabilistic logics, temporal logics, non-distributive quantum logics, and certain intuitionistic logics. It is common to refer to a formal logical system that includes non-truth functional operators as a *non-truth functional logic*.

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

V. MODEL THEORETIC INTERPRETIVE PROPERTIES

By model theoretic or semantic properties, we mean properties that depend on one or more interpretations. In as sense, these are meta-linguistic properties because they relate some aspect of the abstract theory to at least one interpretation. Sometimes, in defining semantic properties, we will refer to a model instead of an interpretation, it being understood that a model necessarily depends on an interpretation.

As noted in earlier articles, by a *permissible interpretation* we mean an interpretation in which any meanings assigned the symbols of the formal logical system are not inconsistent with their standard meanings; that is, their meanings as given in the system's *intended interpretation*. A ***model of a wff*** is defined by a *permissible interpretation* for which the wff evaluates to /𝓣/.

For our purposes, concepts such as truth and interpretation have both syntactic and semantic analogues that are distinct. Truth indicators in the formal language of the Deductive Subsystem are related to, but distinct from, truth values in the Interpretation Subsystem. An interpretation is defined as a set of meaning assignments and a set of truth value assignments specified as rules of correspondence. To be precise, we should always understand a truth indicator or a truth value to be relative to a specific interpretation (or at least the class of interpretations they imply). Then and only then we can say that a wff is "true" (or "false") under that interpretation (or that class of interpretations).

Model theory uses *semantic operators* belonging to the Evaluation Language (e.g., truth functions given as truth tables) to investigate the effect of *permissible interpretations* on a formal logical system. The computation of possible truth values for a wff (e.g., its truth tables), *truth functional equivalence* among wffs, the *semantic validity* of particular wff, *truth functional completeness*, and substitution all depend on the semantic operators defined for the system and how they are defined. Nonetheless, we have attempted give below formal definitions of these concepts that are phrased to be applicable to most systems of interest.

For some of the definitions of properties defined here require at least one interpretation, but are not dependent on the characteristics of any interpretation. Instead, the property must apply to all possible *interpretations* (see the next section for a formal definition of *interpretation*). Thus, although the property is defined in the light of semantics, the interpretive step plays no determining role. Recall that an interpretation of a theory (i.e., an otherwise uninterpreted – and so abstract – formal logical system) is expressed via rules of correspondence, establishing the subject as a model of that theory. When we are concerned with properties applicable to all permissible interpretations, the resulting semantic properties are then – in a sense – inherent in the formal logical system. It is not uncommon for such properties to be considered *syntactic*. In this series, because we wish to emphasize formal interpretation and so are not particularly concerned with *logical calculi*, we identify them as semantic in character.

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

Logical Calculi and Interpretations

Concepts pertaining to interpretations require reconsideration when the formal logical system is a calculus. The distinction between logical operators and semantic operators disappears, the semantic operator being taken as a definition of the logical operator that must be consistent with axioms and rules of inference. Similarly, the distinction between logical truth indicators and the (semantic) truth values to which they correspond are merged both conceptually and mechanically, so that any semantic significance occurs solely within the Deductive Subsystem.

In a logical calculi, only truth valued wffs may be interpreted in the sense of being assigned a truth value. A wff is truth valued if and only if any meaning assignment to a variable can have no affect on its assigned truth value. The specific rules that a wff must satisfy to meet this condition depend on how the logical calculus is defined and such wffs may be given a special name. (For example, in a predicate calculus, a truth valued wff is identified as a "sentence" or "closed wff".)

An ***interpretation of a logical calculi*** is then a particular assignment of truth indicators (and so truth values to which they are equated) to truth valued wffs, no meaning assignments being necessary. A model results from an interpretation in which inferential consequents are true as a logical consequence of the truth indicator assignments to inferential premises dictated by that interpretation.

Satisfiability and Semantic Validity

Satisfiability and semantic validity are fundamental concepts. Although we will provide definitions in this subsection, the reader is warned that there are no universal definitions for these properties: What constitutes satisfiability and what constitutes validity depends on the formal logical system to which they are being applied. Nonetheless, we provide definitions that are as general as possible so that the reader may have some sense of them and have a reasonable chance of arriving at an appropriate definition in a particular context.

We say that a wff is ***satisfied*** by an interpretation if it evaluates to some *designated truth value* /$\mathcal{T}$/ under that interpretation. Inasmuch as an interpretation makes meaning assignments to the components of a wff (symbols of various types) from among the objects belonging to a subject system, what is meant by requiring that wff evaluate to /$\mathcal{T}$/ can vary accordingly. (Alternatively, if the system is a calculus, then the meaning of /$\mathcal{T}$/ is abstract.) Note that some truth valued components of a wff (e.g., ***n***-ary predicates and functions) require multiple meaning assignments before a truth value can be determined or alternatively, that a truth value assignment implies multiple meaning assignments. With the foregoing understanding in mind, a wff is ***satisfiable*** if and only if it is satisfied by at least one interpretation.

Often we will want to know that an interpretation satisfies a set ***S*** of wffs, in which case we say

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

require that those wffs be ***simultaneously satisfiable*** by the interpretation. A wff is ***unsatisfiable*** if and only if it is not satisfied by any interpretation.

A wff is ***falsifiable*** (a.k.a. *refutable*) if and only if it is possible to conceive an argument that could result in the wff evaluating to /***ℱ***/ *under some interpretation*. A *falsifiable wff* is not a false wff, nor is it in any sense incorrect: it just will evaluate to /***ℱ***/ under at least one interpretation.

Tautology-like

A wff is ***tautology-like*** if the result of the evaluation procedure is /***𝒯***/ for every permissible interpretation. Such a wff is a ***tautology*** for those special cases in which the formal logical system is defined to have precisely two truth-values. If the system is a logical calculus, an equivalent definition is that the wff evaluates to /***𝒯***/ under every combination of truth assignments to its truth valued terms.

Every *tautology-like* wff is necessarily *semantically valid*. A *semantically valid* and truth valued wff is necessarily ***tautology-like***. Depending on the formal logical system, the converse of this may not be correct: semantically valid wffs may or may not be tautology-like. In part, this distinction arises because a semantically valid wff need not be truth valued in some systems.[10] A wff that is both *semantically valid* and *truth valued* is necessarily ***tautology-like***. Note that a *tautology-like* wff is sometimes said to be a "*logical" truth* (even though this is a semantic property).

Semantic (Logical) Validity

A wff ***f*** is ***semantically valid*** (a.k.a., *valid with respect to satisfiability, logically valid*, or, sometimes, "*logically true"*) or to have the property of *semantic validity* (a.k.a., *logical validity*) if and only if it is *satisfied* by every permissible interpretation of ***f***. (Take note that it is insufficient for ***f*** to be merely satisfiable for some permissible interpretation.) The wff ***f*** is said to have the property of *semantic validity* (a.k.a., *logical validity*). A semantically valid wff is therefore evaluates to ***/𝒯/*** for every permissible interpretation, even though it may not evaluate to ***/𝒯/*** for every possible truth assignment. In the propositional logic, semantically valid wffs do evaluate to ***/𝒯/*** for every truth assignment and are called *tautologies*. Every permissible interpretation of a semantically valid wff provides a model of that wff. If the negation of ***f*** has no model, then ***f*** is semantically valid. Notice that this definition involves no aspect of the Deductive Subsystem. Notice also that he property of *semantic validity* is restricted to truth valued (e.g., closed) wffs.

[10] For example, in first order predicate logic, (∀x)Px → (∃x)Px is *semantically valid*, but is not a tautology.

Many authors confusingly use the term ***logically valid*** (or ***logical validity***) in place of *semantically valid* (or semantic validity, respectively) as applied to wffs. Readers are warned not to let this confusing use of the adjectives logically and logical erroneously convey a purely syntactic property as when used in terms such as *logical consequence* or *logically valid inference* (defined above). One possible reason for this confusing usage is that, for certain systems, syntactic validity and semantic validity coincide or are congruent. The literature pertaining to such systems often fails to make the congruence of syntactic validity and semantic validity explicit, expecting readers to understand both the fact and the implications.

If a wff is satisfiable for some interpretations and not for other interpretations (i.e., its satisfiability is contingent upon interpretation), we say the wff is ***contingent***. A wff is *contingent* if and only if it is both *satisfiable and falsifiable*. Equivalently, a contingent wff is neither *semantically valid* (satisfied for every interpretation) nor *unsatisfiable*.

An inference is ***semantically valid inference*** if and only if, for every *permissible interpretation* of the premises and the conclusion of some inference, it is impossible for the premises to evaluate to **/𝓣/** and the conclusion to evaluate to **/𝓕/**. Put differentially, a deduction is a *semantically valid inference* if all and only those interpretations that make the premises /𝓣/ also make the consequent /𝓣/. Sometimes, in a semantic extension of the notion of *syntactically valid wff* (defined above), a wff is said to be ***valid*** in a more general sense if it is derivable from the set of all non-logical axioms[11] of the system.[12]

The definition of *semantically valid inference* is almost, but not quite, interpretation independent. With one exception, it is possible to find examples of inferences for all combinations of semantic validity and semantic invalidity versus assignments that make the premises and the conclusion evaluate to some combination of **/𝓣/** and **/𝓕/**. That sole exception is an assignment which makes the premises evaluate to **/𝓕/** and the conclusion evaluate to **/𝓣/**. There can be no example given such an assignment of an inference, as existence of this one situation violates the very definition of a *semantically valid inference*. The foregoing is fundamental to an understanding of the difference between notions of truth and semantic validity, which is in turn crucial to studying formal logical systems.

A semantically valid rule of inference (or simply a semantically valid inference) is a rule of inference that preserves semantic validity: if the premises are semantically valid, then the consequent is semantically valid.

By definition, every axiom of a system is defined to be syntactically valid and the system's rules of inference preserve *syntactic validity*. If the rules of inference also preserve *semantic validity*,

---

[11] Recall that a ***non-logical axiom*** is an axiom that is not satisfied in at least one permissible interpretation, as contrasted with a ***logical axiom*** which must be satisfied in every permissible interpretation.
[12] See Tarski, A, Mostowski, A., Robinson, R. M. Undecidable Theories. ©1953, Dover Publications. Mineola, New York. pp. 6-12, 18.

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

we sometimes say that, given a set of wffs ***S***, the *semantic validity* of a wff ***f*** (or possible multiple wffs) follows from ***S*** as a *logical consequence* – emphasizing the requirement of preserving *semantic validity*, rather than that ***f*** is *satisfied under all interpretations*. We may also say that, if ***f*** is provable from ***S*** (in symbols, $\boldsymbol{S \vdash f}$), then ***f*** is *semantically valid* if and only if each wff of ***S*** is *semantically valid*. Equivalently, we may then also say that ***f*** is a *logical consequence* of ***S*** or there is a formal proof of ***f*** given ***S***.

[**Exercise**: Create an array of inferences covering all possible truth assignments to premises, all possible truth assignments to the conclusion, and all possible validities of the inference (valid versus invalid). You can use English declarative sentences for the premises and conclusion in each case.]

Semantic Consequence and Semantic Entailment

A wff ***f*** is a ***semantic consequence*** of a set of wffs ***S*** (written "$\boldsymbol{S \vDash f}$") if and only if there is no permissible interpretation for which all members of ***S*** evaluate to /$\mathcal{T}$/ and ***f*** evaluates to /$\mathcal{F}$/. Alternatively, we may say that a set of wffs ***S semantically entails*** or ***models*** a wff ***f*** if and only if every permissible interpretation of the formal logical system that *satisfies* ***S*** also satisfies ***f***. In other words, the set of permissible interpretations that make all members of ***S*** evaluate to /$\mathcal{T}$/ is a subset of the set of permissible interpretations that make ***f*** evaluate to /$\mathcal{T}$/.

Model-theoretic Consistency

A wff that evaluates to /$\mathcal{T}$/ for least one interpretation is said to be ***m-consistent*** (a.k.a. ***model-theoretic consistent*** or ***semantically consistent***). If every wff in a set of wffs evaluates to /$\mathcal{T}$/ for some interpretation, it is called a ***model-theoretic consistent set of wffs***. A wff (or set of wffs) is ***m-inconsistent*** (a.k.a. *model-theoretic inconsistent* or *semantically inconsistent*) if it has is no interpretation. Notice that *m-consistency* is the same as *satisfiability*. We present it here as a distinct term in contrast with its proof theoretic counterpart *p-consistency*.

Expressive Power and Expressive Completeness

All uninterpreted formal systems (i.e., an abstract theory), including uninterpreted formal logical systems, have ***expressive power***, defined as the set of subject systems for which some interpretation (a set of rules of correspondence) of the theory exists. Not all formal logical systems have the same *expressive power*. A formal logical system is ***expressively complete*** (or enjoys ***expressive completeness***) with respect to some interpretation if all statements in the subject system's subject language can be expressed in the object language. Note that, if the subject has no subject language per se, then the interpretation effectively provides or "induces" such a language.[13] We sometimes say that a formal language is *expressively complete* with

[13] The closest concept to expressive completeness in a logical calculus is truth-functional completeness.

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

respect to some subject, meaning that it can express all the relevant subject matter. In other words, the formal system or language is sufficiently powerful to be able to represent the subject system completely.

If a formal logical system is ***expressively incomplete*** (i.e., not expressively complete) with respect to some subject, there will be elements of the subject system that the formal system cannot represent. A related difficulty occurs when the formal logical system has more expressive power than the subject. Lacking a better term, we will say the system is ***expressively over-complete*** with respect to the subject. There will then be elements in the formal logical system (e.g., constants, variables, axioms, or theorems) that have (and possibly cannot have) no counterpart in the subject.

Both *expressive incompleteness* and *expressive over-completeness* may be encountered in practice, requiring adjustments to be made. When a formal logical system is *expressively incomplete*, it is necessary to find either an extension of the formal logical system that is expressively complete with respect to the subject or else find a proper subset of the subject system for which the formal logical system is expressively complete. When a formal logical system is *expressively over-complete*, one can establish expressive completeness either by finding a fragment[14] (i.e., a subsystem) of the formal logical system that is expressively complete with respect to the subject or else find a superset of the subject system so that all elements of the formal logical system can be interpreted in terms of the subject system.

If, with respect to some subject, a formal logical system is expressively complete without being expressively over-complete, we say the formal logical system is ***expressively minimal*** *with respect to the subject*. If the formal logical system is axiomatically independent (with respect to rules of inference) and the number of axioms is minimal without loss of expressive completeness, we say the formal logical system is ***deductively minimal***.

If two (or more) formal logical systems can be interpreted by exactly the same set of subject systems, they have equal expressive power. Each is then said to have the semantic property of ***expressive equivalence*** with respect to the other(s)[15] or that any pair of such systems are *expressively equivalent*. In a sense, expressive power and expressive equivalence are the semantic counterparts to the syntactic concepts of deductive power and deductive equivalence (both defined in the previous section). They are, however, quite different. Two formal logical systems may be deductively equivalent but not expressively equivalent.

[Exercise: Can two formal logical systems be expressively equivalent without being deductively equivalent? Justify your answer.]

---

[14] The concept of a fragment of a formal logical system is defined formally below.
[15] Of course, two systems that are completely isomorphic are not only deductively equivalent but expressively equivalent.

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

Truth Functional Semantics

A system is ***truth functional*** if the evaluation of a wff can proceed mechanically from the evaluation of its components. A system is said to be ***truth functionally complete*** if, given a set of truth functions (e.g., truth tables) in the evaluation language corresponding to a set of operators in the object language, we can express every possible truth function (or truth tables) by some combination of the given truth functions. As will be seen, the property of truth functional completeness is important for applications in database theory. Two evaluation language expressions (or the wffs from which those expressions derive) are said to be ***truth functionally equivalent*** if they have the same truth function (e.g., truth table).

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

VI. RELATIONSHIPS BETWEEN SYNTAX AND SEMANTICS

The following properties identify special relationships between syntax and semantics, especially various semantic properties of a system's theorems.

Validity with respect to Evaluation

A ***valid evaluation language expression*** (or the wff from which that evaluation language expression derives) evaluates to /$\mathcal{T}$/ (according to the truth functions) for all possible meaning assignments and subsequent truth value assignments to its components.

(Semantic) Soundness

An individual rule of inference can be ***semantically sound inference rule***, meaning that it is *sound with respect to semantic validity*. The rule of inference preserves semantic validity in the sense that, if a premise if semantically valid, any consequent of the rule of inference is also semantically valid. An *inference* (sometimes called an "argument") is a ***semantically sound inference*** if and only if, for every permissible interpretation, the (syllogistic) premises of the inference logically entail the conclusions.

A *formal logical system* is ***semantically sound system*** (i.e., *sound with respect to tautologousness* or *semantic validity*) if and only if every theorem is *semantically valid* with respect to the theory's intrinsic semantics (i.e., its permissible interpretations). In other words, in a *semantically sound system*, all inference rules are *semantically sound* and therefore all theorems of the formal logical system are tautology-like.

(Semantic) Correctness

A formal logical system is said to be ***correct*** if every interpretation (as implied, for example, by a set of truth assignments) that makes every axiom of the system evaluate to /$\mathcal{T}$/ also makes every theorem evaluate to /$\mathcal{T}$/. To put it another way, a formal logical system is correct if all its rules of inference preserve the truth value /$\mathcal{T}$/: Deductions in a correct formal logical system guarantee that if we start from initial wffs that evaluate to /$\mathcal{T}$/, the resulting theorems will evaluate to /$\mathcal{T}$/. Notice that correctness offers no guarantees if the initial wffs evaluate to /$\mathcal{F}$/.

Semantic Consistency

In a semantically correct logical system (that is, one with the property of correctness), if every wff that is a theorem is also tautology-like, logicians say that the system is ***semantically consistent*** (***a.k.a., m-consistent).*** Put differently, a system is *semantically consistent* if and only if it is

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

both *semantically correct* and *semantically sound*.

Note that the relevant set of interpretations (i.e., those for which a wff is tautology-like) need not make the axioms tautology-like. Of course, the theorems of a correct system provable from the axioms are always tautology-like. In a consistent system, every provable wff evaluates to /***𝓣***/ irrespective of truth assignments, even those interpretations (meaning assignments and truth assignments) that make the axioms evaluate to /***𝓕***/.

If a system is not consistent, we say the system is ***inconsistent***. In an inconsistent formal logical system, the deductive relationship among a set of wffs and the evaluations for that set of wffs do not align provability and evaluation to /***𝓣***/ for at least some set of wffs. Generally speaking, higher ordered logics (those beyond first-order) are inconsistent.

The logician's concepts of consistency and inconsistency are related to, but distinct from, the common notion of being inconsistent (that is, contradictory). Logicians call our common notion ***negation inconsistency***, which is a syntactic property (defined under proof theoretic properties above).

Completeness: Deductive, Refutation, and Semantic

A formal logical system is said to be ***deductively complete*** (a.k.a., *complete with respect to provability*) if (*i*) it is *semantically correct* (i.e., its rules of inference are truth preserving – see definition below) and (*ii*) every tautology-like wff (a semantic property of wffs) is provable (a syntactic property of wffs). In other words, under every permissible interpretation that makes the axioms of the system evaluate to /***𝓣***/, the corresponding subject system's set of expressions that evaluate to /***𝓣***/ will be identical to the set of provable wffs (theorems). From the perspective of models, every true expression in a model of the system is then expressible (according to some interpretation and set of truth value assignments) as a wff that is not only (a) satisfied, but (b) provable.

Whenever a formal logical system is both *deductively complete* and *decidable with respect to provability* (defined above), then an effective procedure must exist to determine whether or not any arbitrary wff is tautology-like. This means that deductive completeness can then be understood in terms of an effective procedure for evaluation (applied over all permissible interpretations) rather than as an effective procedure for deduction (i.e., provability) in the object language). This observation has profound implications for database theory and practice, in particular for the desired properties of formal logical systems that can serve as a foundation for a DBMS.

A system is ***strongly complete*** (a.k.a., "***deductively complete in a strong sense***", ***syntactically complete***, or ***maximally complete***) if no wff can be added to its axiom set that would be independent of the other axioms. For every set of wffs ***S*** in a *strongly complete* system, if a wff ***f*** is a semantic consequence of ***S,*** then ***f*** is provable (i.e., deducible) from ***S***. In other words, if ***S***

semantically entails ***f****,* then ***S*** *logically entails* ***f***. Again, notice the relationship between syntactic and semantic properties.

If a system is *strongly complete* and *p-consistent* (a.k.a., *negation consistent*), it is provable that any new independent axiom can only make the system *inconsistent* (assuming it is to remain *strongly complete*). In another sense, since adding a new axiom to an inconsistent system cannot make that system consistent, a formal system is *strongly complete* if and only if no unprovable sentence can be added to it without introducing an inconsistency.

[**Exercise:** Give an informal proof of the foregoing properties of *strongly complete* and *p-consistent* formal logical systems.]

The property of being *strongly complete* will be an important goal when we discuss applications of logic to database theory. Ideally, we will want to designate a set of wffs that represent the subject as axioms and to know that no others are either necessary or advantageous.

A formal logical system is ***refutation complete*** if for every set of *unsatisfiable* wffs ***S*** that logically entail ***f*** it is possibly to show that ***f*** evaluates to /***F***/. Note the difference between *refutation completeness* (a semantic property) and *negation completeness* (a syntactic property).

If a formal logical system is both (i) *deductively complete* and (ii) all *semantically valid* wffs are theorems (*syntactically valid*)[16], then we say the system is ***semantically complete***. Systems that are *deductively complete* are not necessarily *semantically complete* for the same reasons that all *semantically valid* wffs are not necessarily *tautology-like*. The phrase "*complete with respect to tautologousness*" is sometimes used for the property of *semantic completeness*. ***Semantic completeness*** is the converse of *semantic soundness* for formal logical systems. (Both propositional logic and first order predicate logic are semantically complete.)

Decidability (with the respect to Satisfiability)

A formal logical system is said to be ***decidable with respect to satisfiability*** if it is provable that there exists an effective procedure for determining if an arbitrary wff is satisfiable or not. Such a procedure solves ***the decision problem for satisfiability***. Note that this does not necessarily mean that an effective procedure exists for finding any specific satisfaction of any wff. If an effective procedure exists for finding a satisfaction of any wff (i.e., an *effective evaluation procedure*) it follows that the system is decidable with respect to satisfiability. If a system is *semantically correct* (true axioms imply true theorems for every interpretation) and *decidable with respect to provability*, it follows that it is *decidable with respect to satisfiability*.

[16] Put another way, this condition means the system is deductively powerful enough to prove all true wffs of the intended interpretation – it is *deductively complete*.

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

Decidability (with the respect to Semantic Validity)

A formal logical system is said to be ***decidable with respect to (****semantic****) validity***[17] if it is provable that there exists an effective procedure for determining if an arbitrary wff is semantically valid or not. Such a procedure solves ***the decision problem for validity***. Note that this does not necessarily mean that an effective procedure exists for finding a specific proof of validity.

If a system is both *semantically sound* (every theorem true) and *deductively complete*, then a wff is *semantically valid* if and only if it is *syntactically valid* (i.e., a provable wff). It follows that such a system is *decidable for provability* if and only if it is *decidable for validity*: the proof theoretic concept of decidable for provability and the model theoretic concept of decidable for validity coincide. It is common for authors writing about such systems to use the terms interchangeably and use "decidable" without qualification.

[**Exercise**: Prove from the definitions of semantically sound and deductively complete the assertion of the first sentence of the preceding paragraph. Don't forget in your proof that the asserted relationship between *semantic validity* and *syntactic validity* is a biconditional.]

Semantic Decidability

For formal logical systems that have certain model theoretic properties (*semantic correctness* and *consistency*), decidability may sometimes be determined by a semantic test. In an attempt to keep this special case distinct from the general definition, we will call it *semantic decidability*.

A formal logical system that is *semantically correct* and *m-consistent* is ***semantically decidable*** if there exists an effective procedure for determining the truth value – /$\mathcal{T}$/ or /$\mathcal{F}$/ – of any wff. A formal logical system that is not semantically decidable is said to be ***semantically undecidable***. This means that no *effective* evaluation procedure exists, although there may be an evaluation procedure that has less desirable properties (e.g., it may not be guaranteed to terminate in a finite number of steps). If a formal logical system is *semantically correct* and *m-consistent*, but *semantically undecidable* then there exists no effective evaluation procedure that can determine the truth value of each of the system's wffs is /$\mathcal{T}$/ or /$\mathcal{F}$/, even thought those wffs are in fact either /$\mathcal{T}$/ or /$\mathcal{F}$/. If a formal logical system is *semantically correct*, *m-consistent*, and *deductively decidable*, then it is *semantically decidable*. (This property is important in relational database theory as we shall see.)

[17] Here validity specifically means *semantic validity* although this is rarely made specific.

VII. HOW PROPERTIES DEPEND ON SYNTAX AND SEMANTICS

The following chart summarizes how the properties we have defined relate to and depend on details of the Deduction Subsystem (syntax) and of the Interpretation Subsystem (semantics). In many cases, these properties can be seen to define a relationship between syntax and semantics. The chart lists aspects of the formal logical system as columns and a specific property in each row. An "*" in the intersection indicates that the property is dependent on that aspect of the formal logical system. Note that the lexicon and formation rules jointly determine what constitutes a wff, and axioms and rules of inference jointly determine what is or is not a theorem.

The relevant aspects of a formal logical system that influence its properties are the lexicon (L), the set of logical operations (LO), formation rules (FR), axioms (A), rules of inference (RoI), some other deductive/syntactic factor (OD), the intended interpretation (II), all permissible interpretations (PI), at least some specific interpretation (SI), evaluation procedure and language (E), and the set of truth functions (TF), and some other interpretive/semantic factor (OI). Figure 5.1 shows the relationship between syntactic properties and the Deduction Subsystem. Figure 5.2 shows the relationship between semantic properties and the Interpretation Subsystem.

SYNTACTIC PROPERTY	DEDUCTION					
	L	LO	FR	A	RoI	OD
equivalent formation grammars	*	*	*			
equivalent inferential grammars	*	*	*	*	*	
deductive power	*	*	*	*	*	
deductive equivalence	*	*	*	*	*	
syntactically valid inference	*	*	*	*	*	
syntactically valid wff	*	*	*	*	*	
p-consistent	*	*	*	*	*	
absolutely consistent	*	*	*	*	*	
relatively consistent	*	*	*	*	*	*
logical entailment	*	*	*	*	*	
logical consequence	*	*	*	*	*	
negation complete	*	*	*		*	
p-decidable	*	*	*	*	*	*

Figure 5.1: Syntactic Properties of Formal Logical Systems

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

SEMANTIC PROPERTY	DEDUCTION						INTERPRETATION					
	L	LO	FR	A	RoI	OD	II	PI	SI	E	TF	OI
satisfiability	*	*	*	*	*		*	*		*	*	
falsifiability	*	*	*	*	*		*	*	*	*	*	
tautology-like												
semantically valid	*	*	*	*	*		*	*		*	*	
semantically valid inference	*	*	*	*	*		*	*		*	*	
semantic entailment	*	*	*				*	*		*	*	
semantic consequence	*	*	*				*	*		*	*	
expressive power	*	*	*	*	*		*		*	*	*	
expressive completeness	*	*	*	*	*		*		*	*	*	
expressive equivalence												
truth functional semantics							*	*	*	*	*	
truth functional completeness		*					*	*		*	*	
truth functional equivalence		*					*	*		*	*	
valid w.r.t. evaluation	*	*	*	*	*		*	*		*	*	
semantically sound	*	*	*	*	*		*			*	*	
semantically sound inference	*	*	*	*	*			*		*	*	
m-consistent	*	*	*				*	*	*	*	*	
semantic correctness	*	*	*	*	*			*		*	*	
deductive completeness	*	*	*		*			*		*	*	
strongly complete	*	*	*		*			*		*	*	
decidable for satisfiability	*	*	*			*	*	*		*	*	
refutation complete	*	*	*	*	*		*	*		*	*	
semantic completeness	*	*	*		*		*	*		*	*	
decidable for validity	*	*	*	*	*		*	*		*	*	*
semantic decidability	*	*	*	*	*		*	*	*	*	*	*
decidable for satisfiability	*	*	*	*	*		*	*		*	*	*

Figure 5.2: Semantic Properties of Formal Logical Systems

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

VIII. CONCLUSIONS

The properties of a formal logical system matter. Ideally, we would like a formal logical system to have the properties of being provably *semantically correct*, *consistent*, *expressively complete*, *strongly complete*, and *deductively complete*. This combination is an extremely difficult goal to attain, except when the expressive power of the formal logical system is restricted. As we will see in subsequent articles, logicians typically trade off among these properties. Failing to be aware of which of these properties applies to a formal logical system can lead to rather nasty surprises, especially in applied systems.

	Formal Logical System Type					
Property	*Truth Functional Propositional Logic*	*First Order Monadic Predicate Logic*	*First Order Polyadic Logic*	*Higher Order Logics (in general)*	*Untyped λ-Calculus*	*Simply Typed λ-Calculus*
consistent	**Y**	**Y**	**Y**	**N**	**N**	**Y**
semantically complete	**Y**	**Y**	**Y**	**N**	**N**	**Y**
decidable	**Y**	**Y**	**N**	**N**	**N**	**Y**
syntactically complete	**Y**	**N**	**N**	**N**	**Y**	**N**
negation complete	**N**	**N**	**N**	**N**	**?**[18]	**?**
Turing complete	**N**	**N**	**N**	**Y**	**Y**	**N**

Figure 5.3: Properties of Some Important Formal Logical Systems

There are many surprises in store when one starts to tinker with the structural properties of a formal logical system. In anticipation of results explored in future articles in this series on specific formal logical systems and their comparative properties, Figure 5.3 above[19] will give the reader some idea of the trades-off between expressive power and other desirable properties for some common formal logical systems. It is worth noting that if a formal logical system provides a foundation for Turing complete (i.e., computationally complete) systems, that formal logical

[18] Negation completeness is considered meaningless for the λ calculus.

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621

system will be a higher order logic. Higher order logics, including any possible formal logical system underlying Turing complete system are not generally consistent, nor are they decidable, syntactically (deductively) complete, or negation complete.

We conclude this article with a few notes regarding properties of particular formal logical systems. The meta-mathematical properties discussed herein apply to most formal logical systems: a particular system with either have the property or it will not. However, the reader should not be mislead into thinking these are the only properties that differentiate systems. There are additional properties that we have not covered, primarily because they will not have a direct effect on the choice of a formal logical system as the foundation for a DBMS.

Additionally, the reader is warned that particular types of formal logical systems may have properties that differentiate them in more detail. For example, there are sometimes further variations on validity that apply only to certain types of formal logical system. Given the background provided in this article, it should be relatively easy to understand those further specialized and perhaps subtle differentiations. And, once the relationship between these generally applicable properties to DBMSs is understood, we expect the reader will be equipped to determine how those properties affect the capabilities (or impossibility) of a DBMS founded on systems with or without those properties.

Finally, having provided definitions of the terms consistency, completeness, and decidability, we will be able to provide an explanation of the powerful meta-theorems of Lowenheim-Skolem and Godel when we discuss specific logical systems in following articles. Further interrelationships between syntax and semantics will become apparent and will help us understand the impact of these theorems.

www.AlternativeTech.com mcgoveran@AlternativeTech.com

Post Office Box 4699, Deerfield Beach, FL 33442 Telephone: 831/338-4621